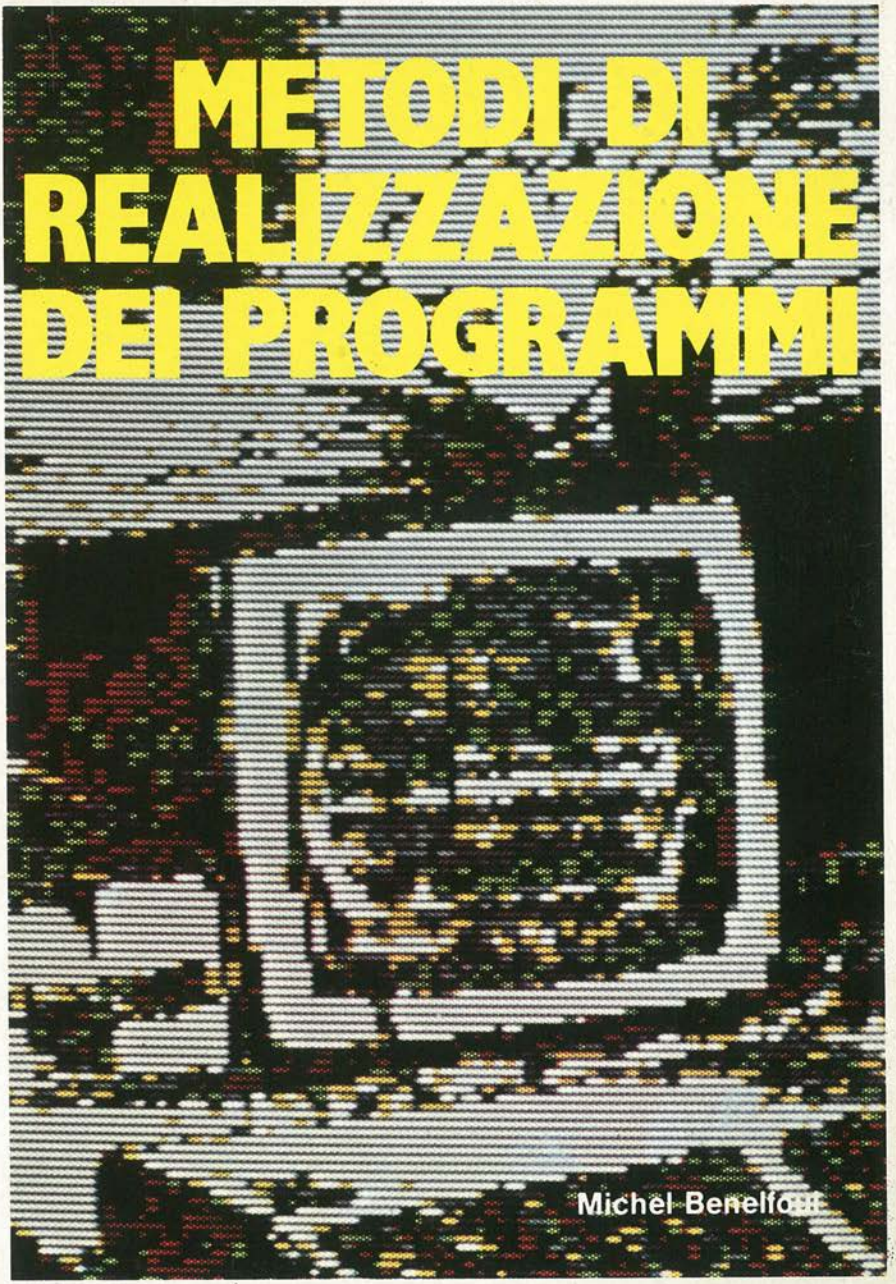
METODI DI REALIZZAZIONE DEI PROGRAMMI



Michel Benelfoul

EDIZIONE ITALIANA

GRUPPO EDITORIALE JACKSON

METODI DI REALIZZAZIONE DEI PROGRAMMI

Michel Benelfoul

GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

© Copyright per l'edizione originale Editions du P.S.I. 1981
© Copyright per l'edizione italiana Gruppo Editoriale Jackson - 1984

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura dell'edizione italiana la signora Francesca Di Fiore e l'ing. Roberto Pancaldi.

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

Fotocomposizione: Lineacomp S.r.l. - Via Rosellini, 12 - 20124 Milano

Stampato in Italia da:
Grafika 78 - Via Trieste, 20 - Pioltello

SOMMARIO

PREFAZIONE V

CAPITOLO 1 - CONCETTI DEL LINGUAGGIO 1

CAPITOLO 2 - CONCETTI DELL'ANALISI 9
- lo studio preliminare 9
- il processo di scelta dell'elaboratore 10
- lo studio tecnico 11

CAPITOLO 3 - LE TAVOLE DI DECISIONE 21

CAPITOLO 4 - METODI DI REALIZZAZIONE DI UN PROGRAMMA 25
- esposizione del problema 25
- elenco dei risultati da ottenere 25
- elenco dei dati d'ingresso 26
- volume dei dati da elaborare 26
- suddivisione dell'elaborazione in blocchi funzionali 26
- i simboli dei diagrammi di flusso 27
- altri esempi di applicazione dei simboli 28

CAPITOLO 5 - IL MANUALE DI PROGRAMMAZIONE 41
- esposizione del problema 41
- maschere di schermo 41
- il diagramma di flusso 41
- identificazione delle variabili 43
- scheda d'identificazione delle variabili 43

CAPITOLO 6 - ESEMPIO APPLICATIVO 47

CAPITOLO 7 - MESSA A PUNTO E MANUTENZIONE 69

APPENDICI

- la teoria degli insiemi 73
- le operazioni booleane 77
- le regole di precedenza nelle operazioni logiche 81
- i diagrammi di Veitch 83
- le tavole di verità 85
- presentazione del listato del programma 87

PREFAZIONE

La realizzazione di un programma richiede metodo e precisione. Le difficoltà incontrate sugli elaboratori personali sono analoghe a quelle affrontate dagli informatici professionisti, a parte le debite proporzioni.

Risulta quindi d'interesse per l'hobbista come per il futuro informatico acquisire fin dall'inizio un metodo rigoroso d'analisi, sostenuto da un supporto di studi e di programmazione adatti.

Questa guida, che vi proponiamo, dovrebbe rispondere ai problemi incontrati utilizzando un elaboratore personale.

Tuttavia non ha la pretesa di essere una bibbia dell'informatica (esistono già diversi lavori sull'argomento e vi raccomandiamo particolarmente le edizioni P.S.I. "COMMENT PROGRAMMER" di Jean-Claude BARBANCE), ma piuttosto di essere un aiuto semplice, facile da consultare e di piccole dimensioni, permettendo di familiarizzarsi rapidamente alla pratica dell'analisi e della programmazione.

CAPITOLO 1

CONCETTI DEL LINGUAGGIO

Nelle discipline tecniche in rapida evoluzione la terminologia non è ben definita e questo comporta una maggiore difficoltà.

L'informazione non sfugge a tale regola e conviene avere a disposizione in ogni momento le rigorose definizioni dei termini utilizzati.

Presentiamo di seguito un breve glossario delle parole, dei termini utilizzati nel contesto dell'informatica applicata.

CONCETTI DEL LINGUAGGIO - TERMINI

Informatica:
Complesso delle tecniche di acquisizione, di ordinamento, di memorizzazione, di trasmissione e di utilizzazione delle informazioni.

Dispositivi periferici:
Settore industriale dell'informatica che include i dispositivi appartenenti a quattro categorie differenti:
— le periferiche dell'elaboratore,
— i terminali collegati a distanza alle unità centrali attraverso le linee di trasmissione dati,
— i dispositivi di acquisizione dati che trasformano l'informazione in una forma assimilabile dall'elaboratore,
— gli elaboratori di piccola dimensione (minielaboratori, microelaboratori, ecc.), che costituiscono in definitiva la parte più importante di tale settore.

Telematica:
Complesso di discipline e di tecniche applicabili nell'uso congiunto dell'informatica e delle telecomunicazioni.

Elaborazione a distanza:
Modo di utilizzo o di funzionamento di un elaboratore le cui sorgenti d'informazio-

ne distanti dall'unità centrale sono ad esso collegate per via telefonica o telegrafica.

Informatica distribuita:
Informatica che utilizza piccoli sistemi (terminali comandati da un minielaboratore) logicamente e fisicamente collegati tra loro, ma geograficamente lontani dall'elaboratore centrale a cui possono essere collegati e con il quale comunicano saltuariamente...

Informatica remota:
Informatica che utilizza piccoli sistemi (terminali comandati da un minielaboratore), geograficamente tra loro distanti e con cui comunicano solo saltuariamente; le decisioni sono completamente centralizzate, o quasi, ed esiste poca dipendenza tra gli elementi dei sistemi.

Elemento:
Parte costitutiva di un insieme. I dati dell'informazione alfabetica e numerica, la documentazione di ingresso e uscita dell'elaboratore, i file per il supporto delle informazioni, i blocchi di elaborazione, la memoria centrale, le unità logiche e aritmetiche, le periferiche.

Insieme:
Collezione di elementi, in numero finito, o indefinito, suscettibili di assumere determinate proprietà (il cui criterio di appartenenza a questa collezione è priva di ambiguità) e di avere tra loro, o con gli elementi di altri insiemi, certe relazioni (N. BOURBAKI, 1939).

Struttura:
Modo di considerare un insieme in funzione delle leggi di composizione in esso definite. Disposizione delle parti d'un insieme astratto, d'un fenomeno o d'un sistema complesso, considerata generalmente come caratteristica di quell'insieme e come durevole.

Sistema:
Insieme coordinato di oggetti tendenti ad ottenere un risultato. Composto da elementi o costituenti, è dotato di una struttura quando: - i suoi costituenti possiedono certe proprietà in quanto appartenenti al sistema; - alcune di queste proprietà sono funzionali, cioè, influenzano l'evoluzione del sistema. Esiste il sistema di elaborazione, il sistema operativo.

Architettura del sistema:
Struttura di un sistema di elaborazione dati geograficamente disperso e costituito principalmente da una o più unità centrali, da unità di trasmissione e da stazioni terminali.

Architettura distribuita:
Struttura di un sistema di elaborazione dati in cui i sottosistemi (terminali collegati a un minielaboratore o a un microelaboratore) sono logicamente e fisicamente collegati tra loro, ma geograficamente lontani dall'elaboratore centrale con cui comunicano solo saltuariamente.

Architettura remota:
Struttura di un sistema di elaborazione dati in cui i sottosistemi (terminali collegati a un minielaboratore o a un microelaboratore) sono geograficamente lontani dall'elaboratore centrale con cui comunicano solo saltuariamente; le decisioni sono completamente decentrate o quasi, con pochissime dipendenze tra gli elementi del sistema.

Analisi:
Scomposizione di un dato problema per rivelarne gli elementi costitutivi e i legami che li uniscono in previsione dell'elaborazione sulla macchina. Un carattere è analitico se considera le cose nel loro elemento (A. LALANDE).
Il carattere analitico deve unirsi a un carattere sintetico.
Insieme, sistema costituito da fenomeni solidali, in modo che "ognuno dipende dagli altri e non può essere quello che è se non attraverso, e a causa, della sua relazione con essi". (A. LALANDE)
Nel realizzare un sistema di elaborazione automatico, l'analisi viene applicata nelle diverse fasi della realizzazione; ciò è sviluppato nel capitolo che tratta i concetti dell'analisi. In informatica termine introdotto dall'uso per delineare:

— i lavori di studio di una applicazione e la progettazione di una nuova soluzione; questi lavori vanno talvolta sotto il nome di analisi dei sistemi ("systems analysis") o di progettazione funzionale ("functional design");

— la definizione precisa delle elaborazioni eseguite del calcolatore e delle procedure esterne che permettono di realizzare questa soluzione (analisi di sviluppo - software analysis - o studio tecnico).

Tavola di decisione:
Tavola di corrispondenza fra tutte le combinazioni delle condizioni che possono influire nella soluzione di un problema e i passi da eseguire in ognuno dei differenti casi.
Nota - l'importanza delle tavole di decisione è di evitare la ripetizione delle condizioni e dei passi comuni a un insieme di tali tavole. (strumento di analisi dell'informatico).

Software:
Insieme dei lavori di logica, di analisi, di programmazione, necessario per il funzionamento di un sistema di elaborazione. Ci limiteremo a spiegarlo in termini di programmazione, per cui esso si compone di due parti:

Il software di base che riunisce i programmi standard del costruttore (di un determinato elaboratore) e messo a disposizione dell'utente: programma di ordinamento, di fusione, di composizione testi, di riversamento di un file su stampante, ecc.
Il software applicativo che riunisce i programmi gestionali progettati e scritti o composti dall'utente per suo uso.

Package applicativi:
Insieme coerente e indipendente di programmi, di servizi, di supporti d'informazione e di documentazione, concepito per realizzare elaborazioni di informatica generale, la cui diffusione riveste un carattere commerciale e che un utente può utilizzare in modo autonomo dopo un'installazione e un addestramento limitato.

Benchmark:
Problema di test proposto da un utente (piccola, media o grande impresa), al fine di esprimere attraverso una gara d'appalto le proprie richieste tecniche a diversi costruttori di elaboratori.
L'analisi dei risultati ottenuti indica l'adattabilità e le caratteristiche dei dispositivi proposti al dato problema.

Microprogramma
(Firmware in inglese). In un elaboratore l'insieme dei programmi registrati permanentemente in una memoria non modificabile durante l'elaborazione, allo scopo di permettere l'esecuzione delle istruzioni attraverso microprogrammi o per eseguire una elaborazione specifica.
Nota - Il microprogramma può anche essere completamente o parzialmente cablato, ma presenta allora l'inconveniente di essere definitivo, al contrario di quanto accade quando è registrato in memoria.
Il microprogramma permette eventualmente di modificare il modo operativo di un elaboratore, per adattarlo alle particolari applicazioni di un utente.

Hardware:
Indica tutto ciò che compone fisicamente un elaboratore: [esempi: la memoria a ferrite, a circuiti integrati, a dischi, la console di comando, le periferiche quali lettori di nastro, unità a dischi, cavi di collegamento ...].

Processore:
Hardware destinato, in un elaboratore, alla esecuzione delle istruzioni. Comprende fondamentalmente gli organi di comando, di calcolo e i registri. Il processore di un elaboratore è costituito dall'organo centrale di comando, i circuiti di calcolo aritmetico e logico e le memorie interne, con esclusione di tutta o parte della memoria principale.

Memoria RAM:
(Random Access Memory- Memoria ad accesso diretto). Memoria il cui contenuto

può essere modificato a volontà da ogni utente autorizzato.
Si contrappone alla memoria ROM.

Memoria ROM:
(Read Only Memory - Memoria di sola lettura). Si contrappone alla memora RAM. Memoria a semiconduttori, il cui contenuto è modificabile elettricamente e in modo selettivo.
Questo termine fa riferimento a una memoria inalterabile; cioè, una memoria in cui nessuna informazione può essere scritta durante l'esecuzione del programma.

Sono utilizzati altri termini, come:

EEPROM (Elettrically Erasable Programmable Read Only Memory); programmabile elettricamente e cancellabile elettricamente in modo selettivo.

EPROM (Erasable Programmable Read Only Memory) ; programmabile elettricamente e cancellabile in modo non selettivo mediante esposizione ai raggi ultravioletti.

Con i termini seguenti affrontiamo da un lato la nozione di informazione, indicando i supporti che servono da veicolo, e dall'altro i supporti che rappresentano l'informazione nei blocchi logici d'elaborazione.

I supporti fisici di ingresso/uscita:

Questo termine fa riferimento ai dispositivi e di solito anche all'unità di lettura /scrittura corrispondente:

— tastiera del terminale (solo come unità di ingresso)
— registratore magnetico
— unità a disco, a dischetto
— lettore di nastro magnetico
— lettore di schede perforate, di nastro perforato (solo come unità di ingresso)
— perforatore di schede (solo come unità di uscita)
— schermo del terminale (alfanumerico, grafico)
— stampante

Terminale locale:
Terminale generalmente asincrono comprendente uno o più organi d'ingresso (tastiera, lettore, unità di acquisizione diretta,...) e/o di uscita (schermo, stampante per caratteri, unità a risposta vocale...) riuniti in un'unica stazione di lavoro.

Il terminale locale è un terminale di dialogo che non effettua di per sè alcuna elaborazione e non dispone di altra logica se non quella necessaria alla registrazione e trasmissione dei dati.

Terminale intelligente:
Terminale sincrono comprendente uno o più organi utilizzabili per gli ingressi/uscite veloci dell'elaborazione a lotti.

Il terminale intelligente è generalmente costruito in logica cablata oppure mediante un microelaboratore; il suo utilizzo richiede un certo numero di periferiche classiche (dischi, nastri magnetici, ecc...).

Questo tipo di terminale viene usato principalmente sia come stazione di trasmissione/ricezione di lotti importanti di dati, sia come unità autonoma capace di realizzare determinate elaborazioni, e quindi di trasferire i risultati verso una stazione centralizzata.

Collegamento sincrono:
a) - Indica un elaboratore o uno dei suoi organi, nel quale l'inizio di ogni operazione, e spesso il ritmo del suo svolgimento, sono deteminati da segnali regolarmente spaziati nel tempo, provenienti da un "clock" (orologio).

b) - Indica un modo di trasmissione dei dati, dove l'istante di trasmissione di ciascun elemento binario ("bit") è determinato da segnali regolarmente spaziati nel tempo.

Collegamento asincrono:
a) - Indica un elaboratore i cui diversi organi funzionano indipendentemente da un solo segnale di sincronizzazione imposto da un orologio. L'inizio di ogni operazione o elemento d'operazione è determinato da un segnale emesso alla fine dell'operazione o elemento d'operazione precedente, o al rilascio degli organi necessari per eseguire l'operazione o elemento d'operazione successiva.

b) - Indica un modo di trasmissione dei dati dove l'inizio della trasmissione di un carattere o di un blocco può accadere in qualunque istante, restando spaziata da un intervallo di tempo la trasmissione dei bit successivi che compongono ciascun carattere.

Stampante ad impatto: (o stampante ad aghi).
Stampante dove ogni carattere viene stampato per punti, seguendo una determinata configurazione in un reticolo regolare di punti (matrice di punti).

Una matrice è definata da un insieme costituito da un numero di colonne e da un numero di righe. La qualità di stampa dipende dal numero di punti della matrice. La velocità di stampa si esprime in caratteri al secondo.

Esempi: 7 * 5
7 * 7
9 * 7
9 * 9 ecc.

Stampante termica:
Stampante a matrice di punti in cui la testina di stampa è costituita da resistenze che impressionano per punti un foglio termosensibile. La velocità di stampa si esprime in caratteri al secondo.

Stampante per caratteri: (o stampante seriale)
Stampante in cui i caratteri di una riga sono battuti l'uno dopo l'altro.

La velocità di stampa si esprime in caratteri al secondo.

Stampante a banda:
Stampante al volo in cui i caratteri di stampa sono battuti su un nastro flessibile senza fine svolgentesi ad una velocità costante davanti alla riga di stampa.

La velocità di stampa si esprime in righe al minuto.

Stampante a laser:
Stampante non ad impatto in cui un raggio luminoso emesso da un laser disegna su una superficie fotosensibile un'immagine latente, sviluppata e fissata in seguito.

Nota: le stampanti non ad impatto non permettono una stampa in multicopia.

I supporti logici di ingresso/uscita:
Questo termine indica il contenuto. Riguarda il file di dati, la cui struttura risponde alla suddivisione basata sui livelli gerarchici seguenti:

— il *blocco*, di uno o più record

— il *record*, di uno o piu campi

— il *campo*, in Basic, è rappresentato da una variabile.
E' la più piccola unità d'informazione indirizzabile da programma.

L'algoritmo:
E' un processo finito e determinato. Si esprime con un numero fisso di direttive, in rigorosa successione. La scelta e l'ordine non devono in nessun caso essere arbitrari.
Un programma può contenere un algoritmo di calcolo, di controllo, di formattazione dei dati.

Sequenza iterativa:
Indica un'elaborazione o una procedura che esegue in modo ripetitivo un gruppo di operazioni, fino al raggiungimento di una determinata condizione.
Un ciclo di programma costituisce una sequenza iterativa.

Il modo operativo:
Il modo di *elaborazione a lotti*, detto *"tempo differito"* in opposizione al termine *"tempo reale"* presentato più avanti; questo modo operativo corrisponde all'elaborazione di un lotto di dati in modo automatico, senza interventi manuali durante l'elaborazione.
Il modo di *elaborazione interattiva* o *conversazionale* detto *"tempo reale"*; consiste in uno scambio istantaneo di informazioni tra l'utente e l'elaboratore. Il dialogo si svolge nella forma di domande/risposte.

Le funzioni automatizzate:
Indicano le operazioni effettuate esclusivamente dall'elaboratore: [esempi: lettura del file, elaborazione dei dati (controllo, calcolo, formattazione...), stampa di documenti].

Le funzioni manuali:
In contrapposizione alle funzioni automatizzate, s'intende tutto ciò che non è compiuto dall'elaboratore.

La transazione:
Messaggi scambiati tra l'operatore (attraverso la tastiera del terminale) e l'elaboratore, per mezzo del programma contenente le domande e le risposte corrispondenti a tutte le possibili eventualità.
In una configurazione che presenta un microelaboratore in linea con un file clienti su disco magnetico, la successione delle operazioni può essere così suddivisa:

- lettura del file clienti,
- selezione,
- modifica conseguente del file clienti.

Una transazione consiste, ad esempio, in una o più modifiche di uno o più record di un file, sulla base di un gruppo di informazioni indicanti le modifiche da compiere.

La sessione:
Elaborazione di un certo numero di transazioni rispetto ad una data unità di tempo.

CAPITOLO 2

CONCETTI DELL'ANALISI

Lo studio di progettazione e di realizzazione di un sistema di elaborazione richede un approccio metodico dei problemi da studiare.

Possiamo distinguere tre parti principali:

1 - lo studio preliminare
2 - il processo di scelta
3 - lo studio tecnico

LO STUDIO PRELIMINARE

Detto anche di analisi del sistema.

E' l'insieme dei processi di studio amministrativi e tecnici che permettono di arrivare alla scelta di una soluzione adatta al problema posto.

Comprende le seguenti fasi:

L'ANALISI QUALITATIVA

S'intende lo studio delle regole, dei lavori, dei documenti e delle strutture.

L'ANALISI QUANTITATIVA

S'intende lo studio dei costi dei dati, dei risultati e dei lavori.

LA RIPROGETTAZIONE

Risultato dello studio di sintesi delle due prime fasi, può comportare la costitu-

zione di nuove strutture, la creazione di un sistema automatizzato, o il rinnovo di quello esistente.

LA VALUTAZIONE TECNICA

Se l'hardware del sistema esiste già, si tratterà di verificare se possiede la capacità e le disponibilità di carico sufficienti per elaborare il problema e definire le eventuali espansioni.

In caso contrario, si devono considerare tre casi:

— le caratteristiche dell'hardware e delle condizioni di lavoro;
— i problemi dei locali inerenti al nuovo sistema e delle incidenze finanziarie;
— quale sarà il nuovo impegno in personale?

IL PROCESSO DI SCELTA DELL'ELABORATORE

A questo livello gli studi portano a prendere i contatti con i fornitori.

Nel corso di questa fase si definiscono gli impegni.

IL MANUALE DELLE SPECIFICHE

Documento di consultazione per i fornitori, deve, in base alle procedure appena create, mettere in evidenza le caratteristiche volute dell'hardware, come le sue finalità.

L'ESAME DELLE OFFERTE

Consiste nell'esame delle proposte, momento importante che richiede uno studio comparativo e dettagliato delle prestazioni mediante criteri di scelta rispondenti agli interessi dell'utente.

IL CONTRATTO DI ACQUISTO E DI MANUTENZIONE

Un documento deve essere prodotto per definire le relazioni tra il costruttore e l'utente.

LO STUDIO TECNICO

Detto anche analisi di sviluppo.

Un sistema di elaborazione deve dare agli utenti ogni garanzia sull'affidabilità dell'informazione da elaborare.

Per prima cosa, si deve cercare di fare programmi affidabili.

Un programma affidabile è in grado di trattare tutte le eventualità (nel linguaggio dell'informatica si usa generalmente l'espressione casi), che possono succedere da eventi prevesti e sopratutto da quelli imprevisti.

Supponiamo una sequenza di programma che tratti l'acquisizione dei dati relativi alla fatturazione di determinati prodotti. I dati sono rappresentati da codici, ciascuno dei quali è associato a un prodotto.
L'affidabilità dell'elaborazione di questa sequenza sarà completa, se il programma prevede:

— il legame di un codice, uno e uno solo, con il prodotto;
— i legami di tutti i codici con tutti i prodotti previsti;
— il trattamento dei codici non riconosciuti;
— il trattamento dell'assenza di codice;
— il trattamento dell'assenza del prodotto.

A partire da questo esempio molto semplice ci accorgiamo subito che la programmazione non è un problema trascurabile. Ma questo è il prezzo per avere un programma che offra le garanzie di una corretta elaborazione. Per ottenerla, esistono degli strumenti che aiutono nella progettazione e nella realizzazione, perfettamente in accordo con le norme della programmazione.

Questi strumenti o concetti appartengono alla matematiche moderne, come:

— la teoria degli insiemi;
— le operazioni booleane;
— i diagrammi di Veitch;
— le tavole di verità.

Alcuni esempi di applicazione pratica di questi concetti sono presentati nel capitolo "esempi applicativi". Seguono le informazioni di ingresso del sistema. A questo scopo esse devono essere classificate secondo criteri di valutazione che non lascino ambiguità nei loro aspetti e contenuti.

In primo luogo, l'informazione deve essere classificata in una delle seguenti categorie:

1- Informazione qualitativa: corrisponde all'indentificazione di un oggetto o di una persona. In generale si tratta del nome, l'indirizzo, la professione, un codice facente riferimento al carattere dell'oggetto o della persona.

2- Informazione quantitativa: corrisponde alla nozione di numero, un importo, una quantità.

Il suo contenuto deve essere definito come segue:

— valore alfabetico: il nome della persona;
— valore alfanumerico: il numero di conto corrente;
— valore numerico: un importo.

I CONTROLLI

La definizione delle informazioni secondo il tipo e il contenuto determina il tipo di controllo da eseguire su ogni variabile prima di considerarle affidabili per il proseguimento dell'elaborazione.

Il controllo di validità

Consiste nell'assicurarsi che una variabile abbia un contenuto strettamente conforme a uno tra quelli aspettati.

Esempio: Nel programma di fatturazione presentato nel capitolo 4, il codice cliente può assumere uno dei valori seguenti: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, corrispondenti a 10 clienti possibili. Ogni altro valore potrebbe essere respinto da un opportuno controllo con la visualizzazione di un messaggio d'errore e con un ritorno alla sequenza d'immissione del codice cliente.

Il controllo di plausibilità

Consiste nell'assicurarsi che il contenuto di una variabile sia compreso tra i limiti prefissati.

Esempio: L'immissione di una data comprende il giorno, il mese e l'anno. Per il giorno, il valore introdotto deve essere compreso tra 01 e 31.

Il controllo di compatibilità

Consiste nell'assicurarsi che più variabili posseggano un contenuto conforme ad una relazione prevista.

Esempio: Nel caso del nostro programma di fatturazione, avremmo potuto inserire un controllo di compatibilità tra le variabili Q1 e Q2; Q1 rappresenta un numero di coppie di sci, Q2 un numero di attacchi, supponendo che Q1 non sia mai superiore a Q2.

I FILE

Definizione: Un file è la rappresentazione di un insieme di dati legati logicamente a un insieme di registrazioni ("record") dello stesso tipo, considerato come un insieme unico.

I file hanno secondo il loro ruolo una definizione particolare.

I file principali o file "permanenti"

Contengono le informazioni fondamentali per l'identificazione di una persona. Questo tipo di file serve come riferimento durante le elaborazioni, il controllo, l'aggiornamento e la consultazione.

Il mantenimento di questo tipo di file è legato al tempo di utilizzo; bisogna prevedere per sicurezza la copia di tale file.

I file "di lavoro" o "scratch file"

Contengono le informazioni che si riferiscono alla creazione, variazione e cancellazione di persone.

Sono congiunti al file principale attraverso le unità di elaborazione, di controllo e aggiornamento. In linea di principio, il mantenimento di questo tipo di file è legato alla periodicità stessa delle elaborazioni.

In altre parole, un file "di lavoro" ne elimina un altro.

I file "intermedi"

Servono di legame tra due unità di elaborazione. Ad esempio, tra due ricerche, se

i criteri di ordinamento sono diversi, l'inserimento di una fase di ordinamento creerà un file intermedio.

La sua durata è molto breve, in generale per il tempo di utilizzo nella unità di elaborazione successiva.

I file "indice"

Vengono usati generalmente per consultazioni momentanee, e contengono solo informazioni di riferimento complementari, non previste nei file principali.

ORGANIZZAZIONE DEI FILE

Accesso sequenziale (su nastro magnetico o disco)

Definizione: accedere a un record qualsiasi contenuto nel file dopo aver letto obbligatoriamente i dati che precedono, in base al criterio di classificazione previsto.

Accesso diretto (solo su disco)

Definizione: accedere a un record qualsiasi contenuto nel file senza essere costretti a leggere i dati che precedono, in base al criterio di classificazione previsto.

Accesso sequenziale con indice (solo su disco)

Permette l'accesso diretto a qualsiasi record, lasciando ogni possibilità di un accesso sequenziale in tutti i record.

Accesso sequenziale concatenato (solo su disco)

Ogni record contiene un campo indirizzo che punta al record successivo.

La ricerca binaria o dicotomica (solo su disco)

La ricerca viene effettuata dividendo l'insieme ordinato dei record in due parti, una delle quali viene abbandonata; l'operazione si ripete sulla parte rimanente fino al termine della ricerca.

LE UNITA' DI ELABORAZIONE

Ogni unità di elaborazione è un insieme di funzioni in successione; il tutto rappresenta il programma.

Definire una unità di elaborazione, significa stabilire con precisione le funzioni di acquisizione, di controllo, di calcolo, di stampa degli aggiornamenti, definire la forma e il contenuto delle informazioni immesse e prodotte attraverso i supporti logici d'ingresso e di uscita e i documenti stampati.

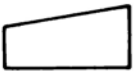
Un sistema di elaborazione può essere formato da un numero di unità ognuna delle quali rappresenta una specifica funzione del sistema.

Esempi
— Unità di acquisizione dei dati in modo interattivo;
— Unità di controllo dei dati;
— Unità di calcolo;
— Unità di aggiornamento del file principale;
— Unità di stampa dei dati convalidati.

SIMBOLI FUNZIONALI

Per rappresentare funzionalmente le unità di elaborazione, così come la loro successione, sono stati definiti un insieme di simboli normalizzati. I simboli sono i seguenti:

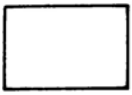
Operazione manuale



Ingresso manuale da tastiera

Linea di collegamento. Trasferimento dell'informazione

Trasferimento dell'informazione per telecomunicazione



Operazione effettuata con l'elaboratore

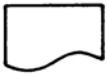
Fusione di due file ordinati con gli stessi criteri di classificazione.

Selezione, in base a criteri diversi, di uno o più elementi, a partire da una sola successione di campi

Ordinamento. Disposizione sequenziale di una successione di campi, secondo un certo criterio

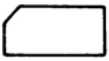
Connettore

Commento



Uscita su stampante - documento

Uscita visiva - Schermo del microelaboratore



Scheda perforata

Nastro magnetico o cassetta

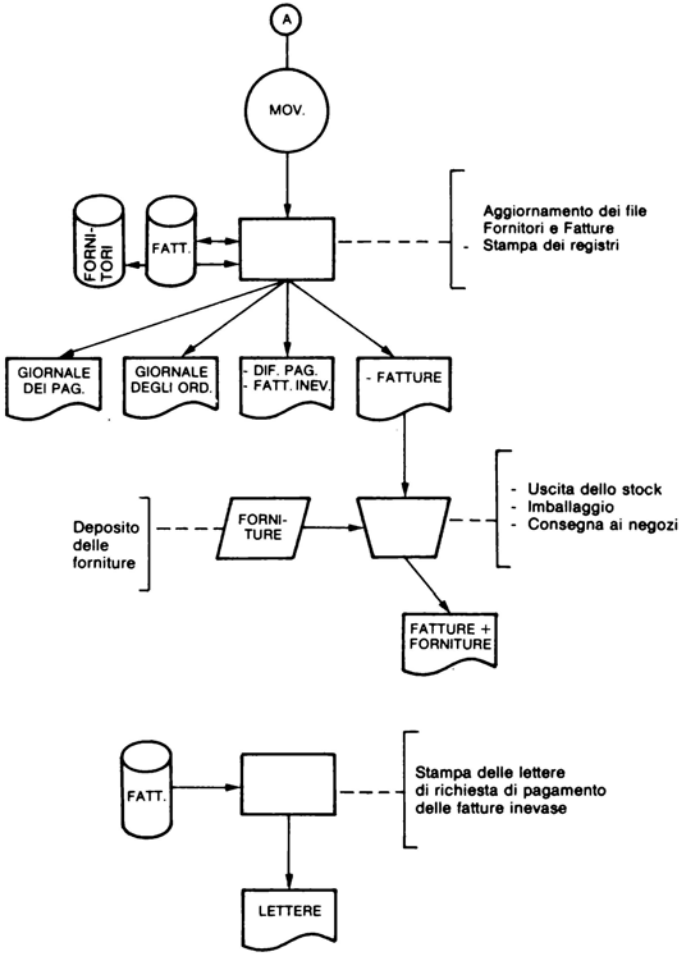
Disco magnetico

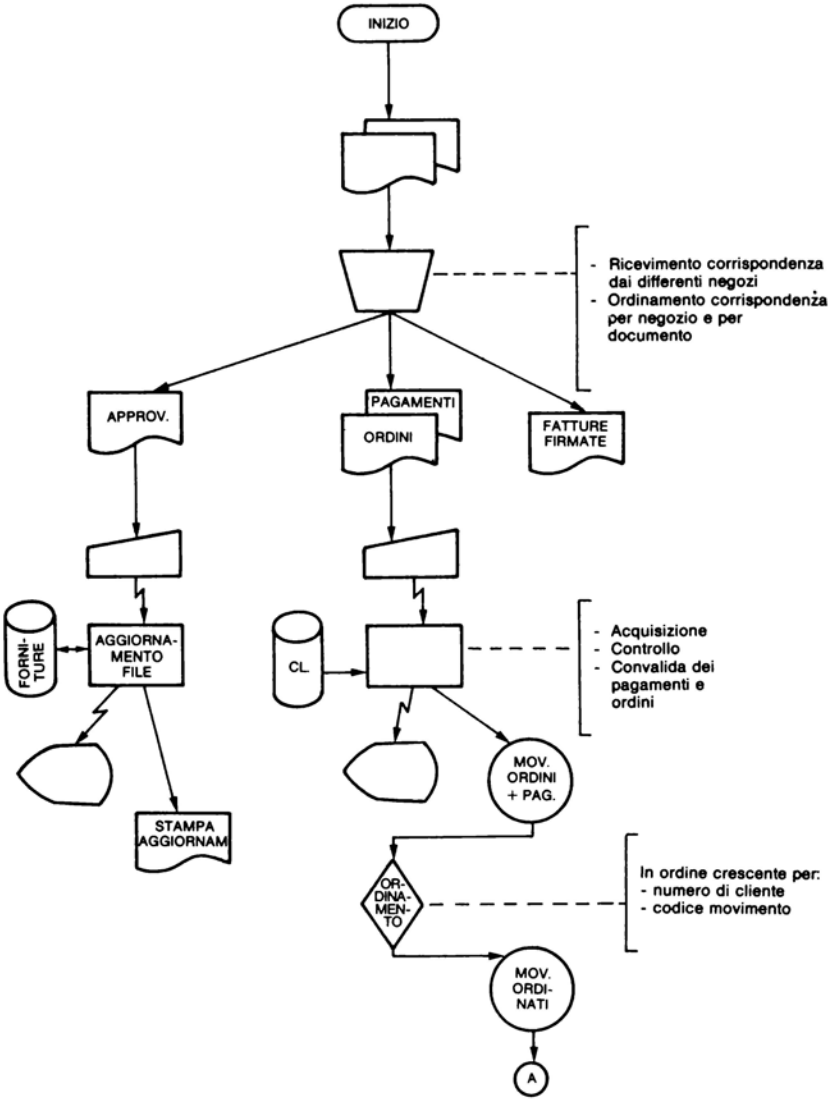
Per spiegare la scelta dei simboli di un diagramma di flusso, secondo le specifiche della AFNR (Associazione Francese di Normalizzazione), abbiamo rappresentato il sistema di gestione di un'azienda, prendendo in considerazione gli ordini, la fatturazione e la consegna di materiale sciistico, presso un certo numero di punti di vendita.

Il sistema di gestione comprende:
— un file delle forniture
— un file clienti
— un file fatture

I movimenti d'ingresso sono:
— i movimenti di riapprovigionamento del file forniture
— gli ordini
— i pagamenti
— le fatture firmate dai clienti

I movimenti di uscita comprendono:
— le fatture
— il registro dei pagamenti
— il registro degli ordini
— le richieste di pagamento delle fatture non evase e le differenze di pagamento
— i tabulati di aggiornamento dei file


A
MOV.
FORNI-TORI
FATT.
Aggiornamento dei file Fornitori e Fatture
- Stampa dei registri
GIORNALE DEI PAG.
GIORNALE DEGLI ORD.
- DIF. PAG.
- FATT. INEV.
- FATTURE
Deposito delle forniture
FORNI-TURE
- Uscita dello stock
- Imballaggio
- Consegna ai negozi
FATTURE + FORNITURE
FATT.
Stampa delle lettere di richiesta di pagamento delle fatture inevase
LETTERE


INIZIO
- Ricevimento corrispondenza dai differenti negozi
- Ordinamento corrispondenza per negozio e per documento
APPROV.
PAGAMENTI
ORDINI
FATTURE FIRMATE
FORNI-TURE
AGGIORNA-MENTO FILE
CL
- Acquisizione
- Controllo
- Convalida dei pagamenti e ordini
MOV. ORDINI + PAG.
STAMPA AGGIORNAM
OR-DINA-MEN-TO
In ordine crescente per:
- numero di cliente
- codice movimento
MOV. ORDI-NATI
A

CAPITOLO 3

LE TAVOLE DI DECISIONE

La realizzazione di un programma richiede il rispetto di determinate regole praticamente costanti da un programma all'altro.

Uno sviluppo preciso è dato nelle pagine che seguono, ma conviene anticipare qualche utile raccomandazione.

Definire l'ambiente del programma, ossia, conoscere bene l'hardaware previsto per il supporto.

Porsi preventivamente la seguente domanda: "Il problema presentato è compatibile per la sua realizzazione con l'hardware prescelto?".

Il tipo di soluzione deve essere determinato cominciando dalla fine, cioè, dai risultati da raggiungere.

Eseguito l'esame completo dei risultati, si costruisce una tabella di equivalenza con i dati disponibili all'ingresso del programma. In tale momento l'utilizzo delle tavole di decisione si rende molto utile.

Il percorso è talvolta complesso tra i dati di ingresso (il prodotto grezzo) e i risultati di uscita (il prodotto finito).

Una tavola di decisione è la rappresentazione schematica sotto forma di tabella di una serie particolare di:

— condizioni (i dati)
— azioni (le operazioni da eseguire)
— regole (operazioni dipendenti da determinate condizioni)

	Regole
Enunciato delle condizioni	Realizzazione delle condizioni
Enunciato delle azioni	Realizzazione delle azioni

Ci sono tre tipi di tavole

— le tavole con voci limitate
— le tavole con voci estese
— le tavole con voci miste

LE TAVOLE CON VOCI LIMITATE

L'enunciato delle condizioni definisce sia una variabile che il suo valore, e l'enunciato delle azioni definisce completamente le azioni. La realizzazione delle condizioni si esprime semplicemente con le parole SI' oppure NO.

Esempio	*R0*	*R1*	*R2*	*R3*	*R4*	*R5*	*R6*	*R7*
Condizione a	sì	sì	sì	no	no	no	no	sì
Condizione b	sì	sì	no	no	sì	sì	no	no
Condizione c	sì	no	no	no	no	sì	sì	si
Azione 1	X							X
Azione 2		X				X		
Azione 3			X				X	
Azione 4					X	X		

Regola 0 = Le condizioni a, b, c sono verificate, si deve eseguire l'azione 1.

Regola 1 = Le condizioni a, b sono verificate, ma non la condizione c, si deve eseguire l'azione 2.

Regola 2 = La condizione a è verificata, si deve eseguire l'azione 3.

Regola 3 = Nessuna condizione è verificata, non si deve eseguire alcuna azione.

Regola 4 = La condizione b è la sola verificata, si deve eseguire l'azione 4.

Regola 5 = Le condizioni b e c sono verificate ma non la condizione a, si devono eseguire le azioni 2 e 4.

Regola 6 = La condizione c è la sola verificata, si deve eseguire l'azione 3.

Regola 7 = Le condizioni a e c sono verificate ma non la condizione b, si deve eseguire l'azione 1.

Nota: Per n condizioni, si possono ottenere 2^n regole.

LE TAVOLE CON VOCI ESTESE

L'enunciato delle condizioni non definisce il valore di una variabile e l'enunciato delle azioni non definisce completamente l'azione da compiere. Così, si rende necessario aggiungere questi complementi d'informazione per ciascuna delle regole.

Esempio:	*R0*	*R1*	*R2*
Condizione a	$0 < x \leqslant 10$	$10 < x \leqslant 50$	$50 < x < 100$
Condizione b	$y = 1$	$y = 2$	$y = 0$
Azione 1	S1	S2	S2
Azione 2	S3	S4	S5

Regola 0 = Se x è maggiore di 0 e minore o uguale a 10, se y è uguale a 1, si devono eseguire le sequenze S1 e S3.

Regola 1 = Se x è maggiore di 10 e inferiore o uguale a 50, se y è uguale a 2, si devono eseguire le sequenze S2 e S4.

Regola 2 = Se x è maggiore di 50 e minore o uguale a 100, se y è uguale a 0, si devono eseguire le sequenze S2 e S5.

LE TAVOLE CON VOCI MISTE

La tavola con voci miste include parti "limitate" e parti "estese".

Esempio:	R0	R1	R2
Condizione A	$0 < x \leqslant 10$	$10 < x \leqslant 50$	$50 < x \leqslant 100$
Condizione B	sì	sì	no
Azione 1	S1	S1	S4
Azione 2	S2		S6

Regola 0 = Se x è maggiore di 0 e minore o uguale a 10, se è verificata la condizione B , si devono eseguire le sequenze S1 e S2.

Regola 1 = Se x è maggiore di 10 e minore o uguale a 50, se è verificata la condizione B, si deve eseguire la sequenza S1.

Regola 2= Se x è superiore a 50 e inferiore a 100, se la condizione B non è verificata, si devono eseguire le sequenze S4 e S6.

PERCHE' LE TAVOLE DI DECISIONE?

Sintetizzano gli elementi di valutazione (enunciato delle condizioni, regole, enunciato delle azioni) in forma rigorosa e compatibile con il linguaggio di programmazione.

QUANDO SI DEVONO UTILIZZARE?

Quando la realizzazione di un'azione o di una combinazione di azioni esigono l'analisi di un insieme di condizioni complesse prima della loro rappresentazione attraverso il diagramma di flusso. Sono ugualmente utili nella messa a punto del programma, permettendo di verificare che tutti i casi possibili siano ben controllati. (vedere a pag. 69).

CAPITOLO 4

METODO DI REALIZZAZIONE DI UN PROGRAMMA

La realizzazione di un programma passa attraverso un certo numero di tappe da eseguire in un dato ordine cronologico.

ESPOSIZIONE DEL PROBLEMA

— quale è il suo argomento?
— come deve essere elaborato?

Alla prima domanda dobbiamo rispondere con la precisa definizione delle richieste, che possono implicare i seguenti paragrafi:

— la definizione dell'argomento da elaborare;
— il volume delle transazioni;
— la periodicità delle elaborazioni;
— il tempo di consegna dei risultati;
— vantaggi e svantaggi della soluzione.

La seconda domanda richiede diverse risposte. Ogni risposta deve essere oggetto di una analisi organica che metta in risalto gli aspetti tecnici specifici dell'hardware scelto.

ELENCO DEI RISULTATI DA OTTENERE
(dati di uscita)

Sotto quale forma:

— Disegnare la maschera di stampa per i risultati.
— Definire le variabili corrispondenti alle uscite su schermo e/o su stampante.
— Definire i file di uscita su cassetta e/o su dischetto.
— Definire le variabili specifiche dei file.

ELENCO DEI DATI D'INGRESSO

Da tastiera:

— Disegnare le maschere per l'acquisizione dei dati su schermo.
— Definire le variabili corrispondenti.
— Definire i controlli di acquisizione.

Dai file (cassetta e/o dischetto):

con i dati d'ingresso,
con i dati di altri file.
— Descrivere il contenuto.
— Definire le variabili corrispondenti.
— Definire le relazioni logiche dei dati.

VOLUME DEI DATI DA ELABORARE

— Calcolo del numero totale di caratteri da acquisire.
— Calcolo del numero totale di caratteri dei file di ingresso.
— Calcolo del numero totale di caratteri dei file di uscita, su cassetta, su dischetto, su stampante.

SUDDIVISIONE DELL'ELABORAZIONE IN BLOCCHI FUNZIONALI

Disegnare il diagramma di flusso

— Inizio (Inizializzazione delle relative variabili).
— Acquisizione alla tastiera (INPUT, GET...).
— Lettura dei file.
— Elaborazione dei dati (controllo, calcolo, confronto, trasferimento, rimessa al valore iniziale delle variabili...).
— Stampa (su schermo, su stampante).
— Scrittura dei file.
— Fine del programma.

I SIMBOLI DEI DIAGRAMMI DI FLUSSO

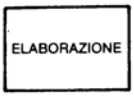
Inizio o Fine di un diagramma di flusso.

Acquisizione di un'informazione da elaborare.

Memorizzazione di un'informazione elaborata.

Operazione che determina parzialmente o completamente il cammino da seguire in una diramazione o in un sottoprogramma. Viene utilizzato per introdurre un simbolo di decisione.

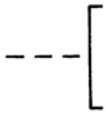
Utilizzazione di condizioni multiple implicanti la scelta di una tra diverse strade.



Operazione o gruppo di operazioni implicanti l'elaborazione di dati per i quali non è definito un simbolo particolare.

Insieme di istruzioni rappresentanti una o più funzioni richiamabili da un qualunque blocco del programma.

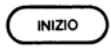
Assicura la continuità dell'elaborazione quando non è possibile rappresentare una linea di collegamento; ad esempio: diagramma di flusso espresso su diverse pagine.



Commento.

ALTRI ESEMPI DI APPLICAZIONE DEI SIMBOLI

INIZIO DEL PROGRAMMA



Un programma bene strutturato si divide come un libro in capitoli, paragrafi, capoversi.

Il capitolo iniziale deve contenere due paragrafi.

P1/L'identificazione del programma - Preambolo

P11 - il titolo
P12 - il nome dell'autore
P13 - la data di realizzazione
P14 - il numero di versione
P15 - la configurazione dell'hardware necessario al funzionamento.

P2/Inizializzazione delle variabili - Introduzione

Questa parte è più critica perchè dipende dalle elaborazioni realizzate nei diversi blocchi dei capitoli successivi.

Ad esempio, l'uso corretto di un indice richiede il valore iniziale 10. Il compito di questo paragrafo consiste quindi nell'assegnare ad esso questo valore prima della sua utilizzazione.

Una variabile è utilizzata come costante; essa contiene un tasso dell'IVA per un dato prodotto. La funzione di inizializzazione attribuisce ad essa nella sequenza di partenza un valore che resterà invariato per tutto il tempo d'esecuzione del programma.

Le variabili usate come accumulatori necessitano di essere azzerate prima di qualunque operazione aritmetica, se non si vogliono avere errori di programmazione (1), o anche errori aritmetici se queste variabili contengono i risultati delle elaborazioni precedenti.

---

(1). Ciò è vero su certi tipi di elaboratori. La somma di valori numerici in un campo "nullo" provoca gravi errori.

La somma aritmetica a più livelli di gerarchia richiede non solo un'inizializzazione delle variabili in uso, ma altresì una rimessa al valore iniziale condizionata, e relativa alle variabili proprie di ciascun livello gerarchico.

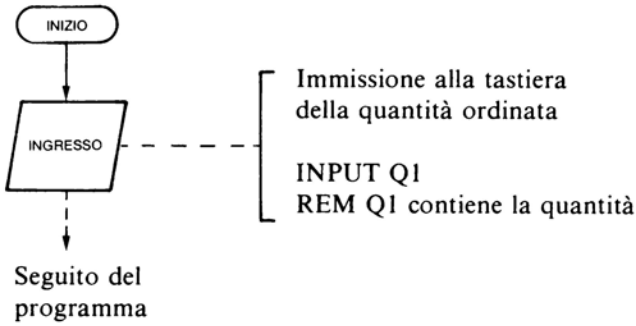
La funzione di rimessa al valore iniziale si effettua nei blocchi di elaborazione presentati più avanti.

INGRESSO

USCITA

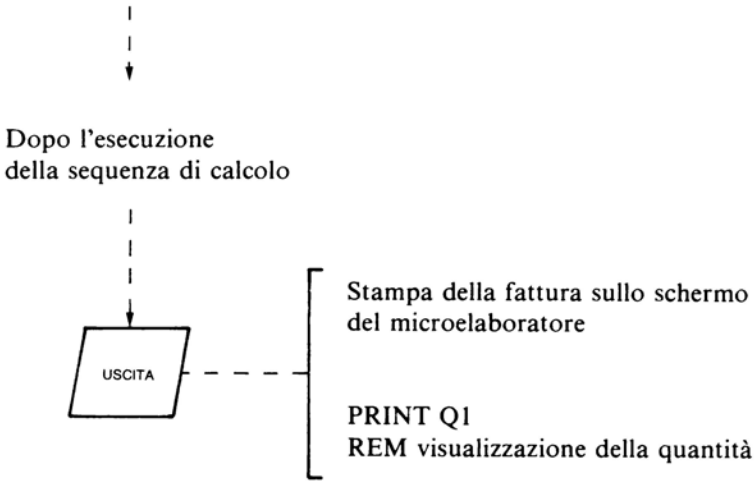
Questo simbolo rappresenta per la *parte ingresso* la funzione di acquisizione di un'informazione o più informazioni alla tastiera, o da un file su dischetto o cassetta.

Esempio: In un programma di fatturazione si richiede l'immissione della quantità ordinata

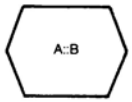


Per la *parte uscita*, il simbolo rappresenta uno schermo, una stampante, o un file su dischetto o cassetta.

Esempio: Dopo la sequenza di calcolo, si procede alla stampa della fattura.



IL SIMBOLO DI PREPARAZIONE



Il simbolo preparazione si traduce in linguaggio di programmazione mediante un'istruzione - IF - che introduce la nozione di confronto prima dell'esecuzione condizionale del blocco di elaborazione.

L'espressione simbolica A::B si legge "confronta A con B". I termini da confrontare devono avere le stesse caratteristiche; ne dipende la correttezza del risultato. Non si confrontano litri con chilogrammi.

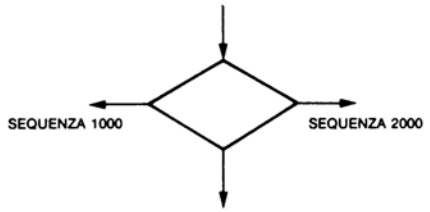
E' quindi molto importante che il contenuto delle variabili da confrontare sia omogeneo; esse possono venire classificate in base a quattro tipi di valori possibili:

- valore puramente numerico (da 0 a 9 esclusivamente);
- valore intero (da 0 a 9, senza cifra dopo la virgola- Basic);
- valore algebrico (da 0 a 9 con il segno + o -); il Basic considera spesso implicitamente i valori numerici con segno;
- valore alfanumerico (combinazione di valori alfabetici, valori numerici, caratteri speciali, spazio, punto, ecc.).

Si rende necesario prevedere nella sequenza di partenza la inizializzazione delle variabili utilizzate nei termini del confronto, dato che spesso i programmatori cadono nella seguente trappola:

- quando si confrontano due termini, considerati puramenti numerici, se uno dei due non è stato inizializzato, cioè riempito con degli zeri, e se non ha assunto durante l'elaborazione valori numerici, contiene solamente il valore - nullo - e spesso il risultato del confronto rinvia ad un salto condizionato opposto a quello aspettato.

IL SIMBOLO DI DECISIONE



Come indica il nome, si tratta di scegliere una strada tra più alternative. Allo stesso modo di un viaggiatore arrivato ad un incrocio, il programma deve prendere una decisione sulla direzione da seguire: a sinistra, a destra, oppure diritto; cioè, andare all'indirizzo 1000, o all'indirizzo 2000 oppure continuare semplicemente la sequenza delle istruzioni successive (equivale, per il viaggiatore, a proseguire diritto).

Per spiegare il ruolo svolto dal simbolo di decisione, ecco un esempio di funzionamento simulato di una stazione di smistamento.

Esposizione del problema

Un posto di smistamento deve selezionare i vagoni in base alla loro destinazione e caratteristiche per guidarli nel loro instradamento.

I criteri di selezione sono:

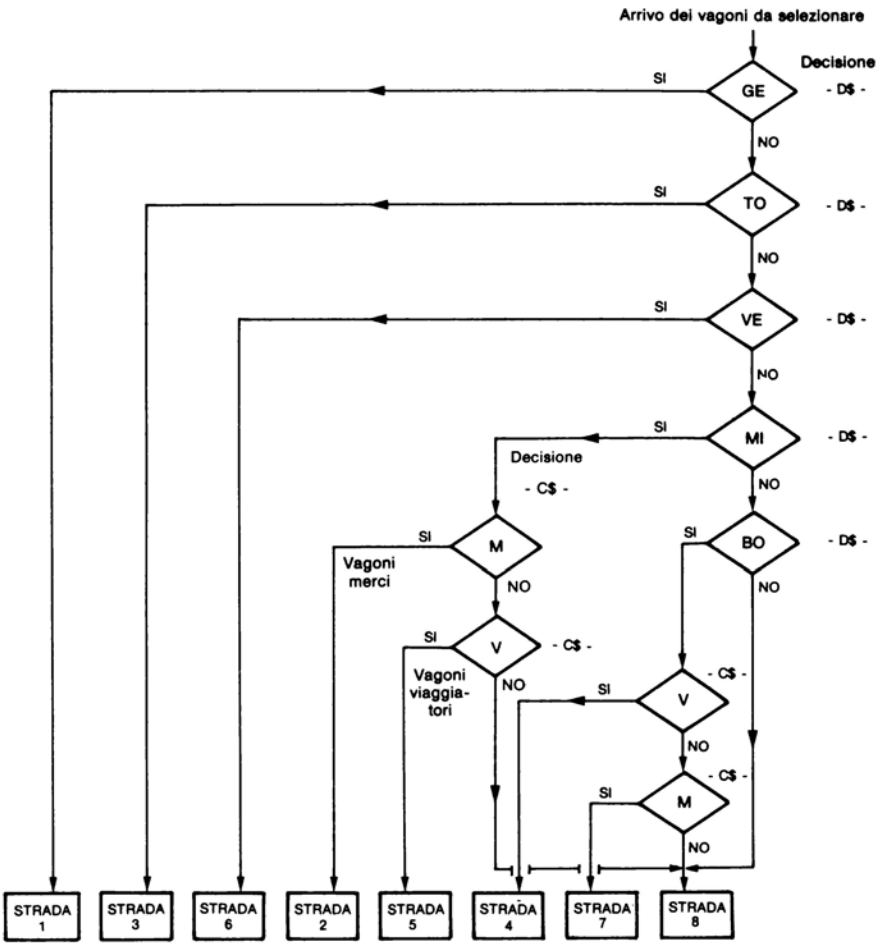
1 - il codice direzione - variabile D$
2 - il codice tipo del vagone - merci - , - viaggiatori - variabile C$.

La tabella di formazione dei treni viene stabilità così:

Direzione	Codice direzione	Codice tipo	Smistamento
GENOVA	GE	M	1
MILANO	MI	M	2
TORINO	TO	V	3
BOLOGNA	BO	V	4
MILANO	MI	V	5
VENEZIA	VE	V	6
BOLOGNA	BO	M	7
Senza destinazione	—	—	8

DATA	NOME DEL PROGRAMMA: **SELEZIONE DEI VAGONI**	PAGINA

DIAGRAMMA DI FLUSSO



La destinazione di alcuni vagoni richiede un solo simbolo di decisione, mentre per gli altri ne sono richiesti due, in particolare per i vagoni con destinazione Milano e Bologna.
La codifica del diagramma di flusso (pagina seguente) diventa a questo punto una formalità e una garanzia di corretto funzionamento, fin dal primo istante!!!

DATA	NOME DEL PROGRAMMA: SELEZIONE DEI VAGONI	PAGINA

FOGLIO DI PROGRAMMAZIONE BASIC

RIGA SCHERMO	NUMERO DI RIGA	NUMERO DI COLONNA SCHERMO
1	100	IF D$="GE"THEN PRINT"STRADA 1"
2	110	IF D$="TO"THEN PRINT"STRADA 3"
3	120	IF D$="VE"THEN PRINT"STRADA 6"
4	130	IF D$="MI"ANDC$="M"THENPRINT"STRADA 2"
5	140	IF D$="MI"ANDC$="V"THENPRINT"STRADA 5"
6	150	IF D$="BO"ANDC$="V"THENPRINT"STRADA 4"
7	160	IF D$="BO"ANDC$="M"THENPRINT"STRADA 7"
8	170	REM ••• L'ISTRUZIONE SEGUENTE INDICA
9	180	REM ••• I VAGONI SENZA DESTINAZIONE
10	190	PRINT"AL DEPOSITO STRADA 8"

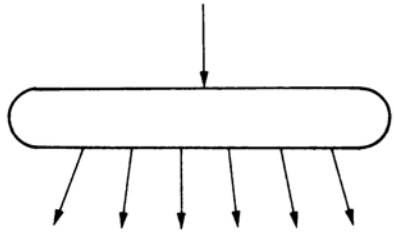
Alcuni linguaggi Basic più evoluti possiedono per i salti condizionati una funzione supplementare (bisognarebbe dire piuttosto di valutazione); si tratta della condizione - ELSE - (altrimenti).

Ossia:

```
IF (Se..) D$ = "MI" AND C$ = "M" THEN (Allora eseguo..)
PRINT "STRADA2" ELSE (ALTRIMENTI eseguo..) PRINT "STRADE"
```

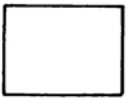
Questa sottile particolarità permette quindi il concatenamento di due istruzioni - IF - in una sola e di andare in due direzioni possibili invece di andare in sequenza (cioè, diritto). In tal caso possono essere utilizzate le tre uscite del rombo.

Esiste ugualmente, in Basic, la possibilità di un unico test con possibilità di rinvii multipli. E' l'istruzione ON ...GOTO ... oppure ON...GOSUB... In questo caso il test del valore di una variabile numerica potrà rinviare su un numero di diramazioni superiore a tre. Sarà allora utilizzato il simbolo seguente:



Queste istruzioni molto potenti hanno necessità, in generale, di una "preparazione" importante. Ad esempio, nel caso dei treni si sarebbe potuto calcolare il numero della strada e fare un unico test di selezione.

ELABORAZIONE



Un blocco di elaborazione rispecchia un'istruzione;

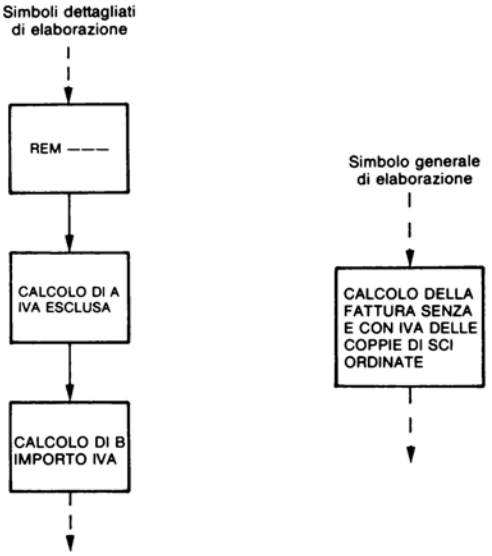
Esempio: Calcolare l'IVA su un totale di prodotti ordinati

```
100 A = Q1 * 18
```

oppure più istruzioni.

Esempio:

```
100 REM Q1 Rappresenta il numero di coppie di sci
110 REM P2 Rappresenta il prezzo unitario di una coppia di sci
120 REM T1 Rappresenta l'IVA
130 REM A Rappresenta il prezzo della fattura, IVA esclusa
140 REM B Rappresenta l'importo IVA della fattura
150 A = Q1 * P2
160 B = A * T1
```



I SOTTOPROGRAMMI

Il sottoprogramma rappresenta un'istruzione o un gruppo di istruzioni inserite in un livello qualunque del programma, a cui si può accedere da qualunque altro livello.

Evita la ripetizione inopportuna d'una funzione di cui si ha interesse e risponde alle necessità particolari dei blocchi di elaborazione nei diversi livelli del programma.

In Basic l'accesso ai sottoprogrammi avviene mediante l'istruzione GOSUB xxxx. Letteralmente significa "vai alla subroutine xxxx". Ciò si traduce in italiano con sottoprogramma.

In quale circostanza il programmatore può pensare di rappresentare un particolare blocco con un sottoprogramma?

Prendiamo l'esempio del calcolo e della composizione di una fattura allo schermo di un microelaboratore, non dotato di un software sufficientemente potente da fornire la centratura automatica degli importi da comporre.

La fattura comprende diverse righe, in ragione di una riga per prodotto diverso. Ogni prodotto è oggetto di un processo di calcolo e il risultato deve essere rappresentato con virgola e due decimali; si deve mettere uno zero a sinistra della parte decimale se il risultato non comporta degli interi. (1)

All'uscita da ogni funzione di calcolo relativa ad ogni prodotto, risulta evidente che la formattazione della variabile contenente il risultato è un processo estremamente complesso e che sarebbe assurdo ripetere la stessa sequenza tante volte quante sono le righe da comporre.

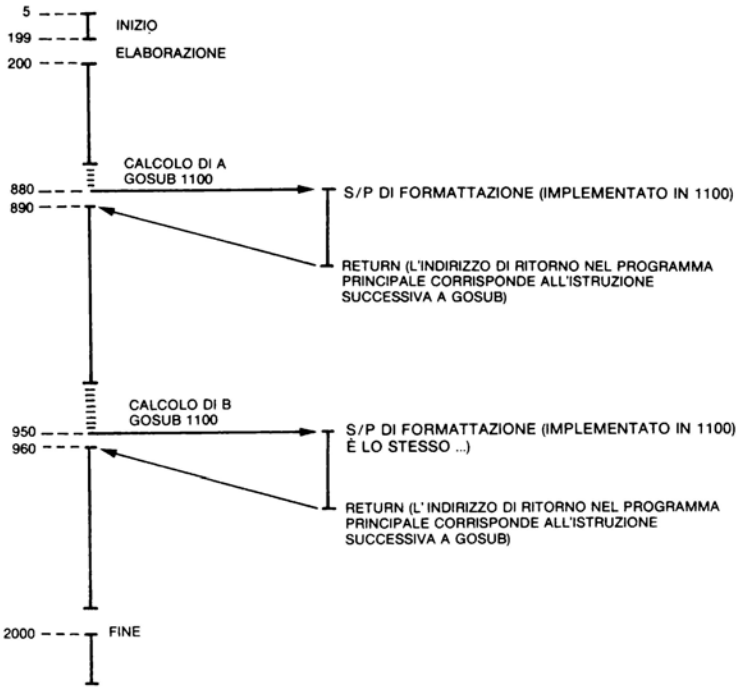
A questo punto interviene la nozione di sottoprogramma che permette al programmatore di guadagnare spazio in memoria e tempo di programmazione.

---

(1) Non preoccupatevi, la soluzione programmata viene data nell'Appendice VI che tratta dell'esempio di realizzazione di un programma.

Il diagramma seguente mostra le relazioni tra le sequenze dette lineari e il sottoprogramma.

DIAGRAMMA DEL PROGRAMMA
con inserimento di un sottoprogramma



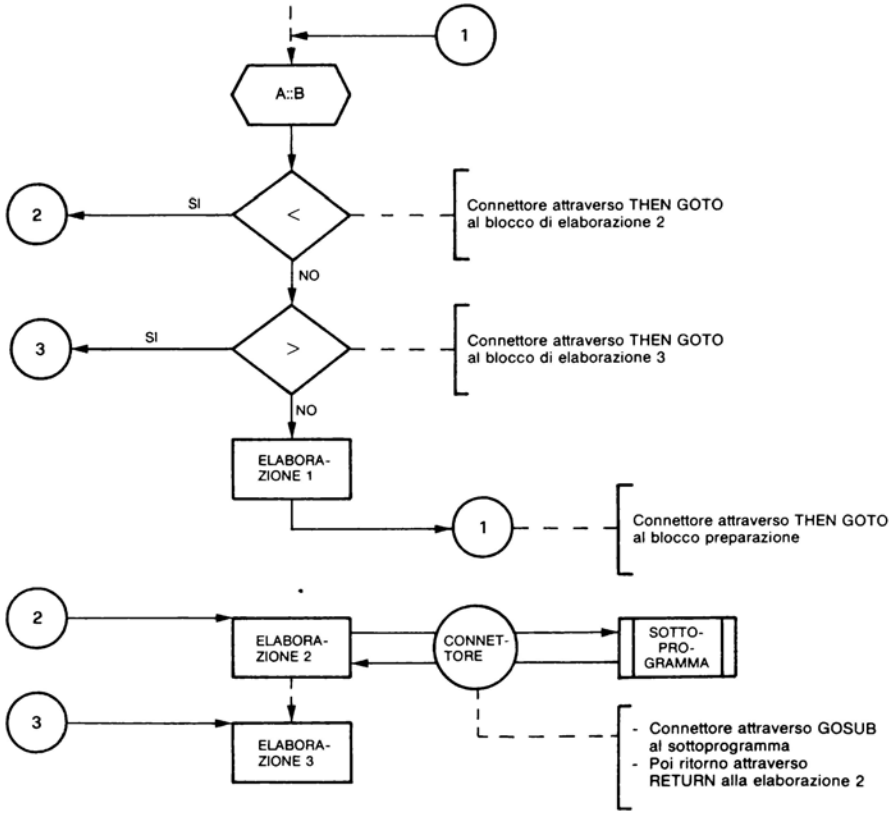
I SALTI DI SEQUENZA

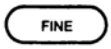
Questo simbolo è utilizzato ogni volta che non è possibile, o non pratico, il collegamento diretto con una linea di diramazione:

1 - GOTO "VAI A" un indirizzo del programma principale saltando una o più istruzioni, sia a monte sia a valle del salto.

2 - GOSUB "VAI A" un indirizzo corrispondente alla prima istruzione di un sottoprogramma.

3 - RETURN "RITORNA A". Ultima istruzione di un sottoprogramma, rimanda nel programma principale all'istruzione che segue immediatamente il GOSUB d'origine.





Con quest'ultimo simbolo siamo all'epilogo della storia, e diciamo, più prosaicamente, alla fine del programma.

Che cosa può contenere questo capitolo?. Ad esempio, un messaggio indicante opportunamente che il programma è finito.

Se si tratta di un programma di gioco, potete pensare alla situazione seguente:

PRINT "LA PARTITA E' TERMINATA - VOLETE RICOMINCIARE?". Se la risposta è negativa, la stampa di un messaggio confermerà l'aspettativa del giocatore.

PRINT "IL PROGRAMMA E' TERMINATO DALL'ISTRUZIONE STOP"

Infatti, l'istruzione Basic di fine programma è - END -.

Il microprocessore, quando incontra questa istruzione, interrompe l'esecuzione del programma e invia il messaggio - READY -.

Un'altra funzione, tipica a fine programma, è quella di comporre una statistica dell'elaborazione che ha senso solo per quelle applicazioni che utilizzano i file.

Le statistiche di elaborazione indicano il numero e l'importo dei dati trattati in ingresso e in uscita del programma. Queste indicazioni posssono servire sia come controllo del corretto funzionamento del programma, sia come valori statistici per necessità particolari.

CAPITOLO 5

IL MANUALE DI PROGRAMMAZIONE

ESPOSIZIONE DEL PROBLEMA

Bisogna sapere naturalmente quale è l'argomento e come risolverlo.

Una breve sintesi deve permettere di impostare chiaramente il problema. La sua realizzazione tecnica ne sarà facilitata.

MASCHERE DI SCHERMO

Disegnare su una griglia di schermo i dati come saranno prodotti dal programma.
Questa fase è molto importante perchè obbliga il programmatore a simulare il lavoro che dovrà essere fatto.

Quando avete stabilito il complesso degli schemi di stampa da ottenere, siete già a metà del vostro lavoro.

Ci sono tre parti da elaborare:
— i dialoghi conversazionali - domande/risposte;
— i risultati (uscite dei file, delle tabelle DATA, dei blocchi di calcolo, ecc.);
— i messaggi dovuti ad errori di programma.

Ogni parte è un sottoinsieme che può essere risolto separatamente. Queste tre parti delineano la suddivisione delle sequenze di programmazione da scrivere in seguito.

IL DIAGRAMMA DI FLUSSO

Permette di concretizzare attraverso i blocchi di elaborazione tutti i compiti che il programmatore deve assumere.

Dà una visione cronologica degli eventi da elaborare, ed è praticamente la radioscopia del futuro programma. Il diagramma di flusso è indispensabile per dare, a priori, una certa coerenza alle sequenze di programmazione.

IDENTIFICAZIONE DELLE VARIABILI

Un programma ha spesso necessità, sopratutto nel campo gestionale, di utilizzare un numero considerevole di campi di informazione, chiamati più correntemente nel mondo dei microelaboratori come variabili.

Ogni variabile deve essere identificata da un nome di battesimo.

A mano a mano che ci si inoltra nel problema, si ha rapidamente un'inflazione del numero delle variabili disponibili; così, per metter ordine, è saggia precauzione farne un sistematico elenco.

La scheda di identificazione presentata più avanti introduce i parametri necessari per identificare ogni variabile.

L'uso ricorsivo di una variabile richiede, nella maggior parte dei casi, l'inizializzazione del suo contenuto. A questo scopo si devono introdurre con accortezza due funzioni importanti di programmazione:

— l'inizializzazione del contenuto;
— la rimessa al valore iniziale del contenuto.

Si può individuare facilmente il blocco di esecuzione di una inizializzazione; è molto più difficile, invece, individuare una rimessa al valore iniziale, ed è ancora meno evidente determinare, tra le variabili in uso, quelle che devono essere inizializzate e ricercare in esse quelle che devono essere riinizializzate.

Una tabella sinottica delle variabili, presentata più avanti, fornisce i criteri per operare con le due funzioni, a seconda del tipo della variabile.

SCHEDA D'IDENTIFICAZIONE DELLE VARIABILI

NOME SIMBOLICO

Il nome contiene in sè il carattere della variabile:
A numerica (intero + decimale)
A% numerica (solo intero)
A$ alfanumerica (stringa di caratteri).

TIPO DI VARIABILE

Decisione: valore relativo indicante almeno due stati possibili

Contatore: incrementato da operatori aritmetici o da funzioni matematiche

Costante: valore posto all'inizio del programma in un determinato stato iniziale e che rimane inalterato per tutto il tempo di esecuzione del programma

Puntatore: contiene l'indirizzo relativo ad un dato definito in una lista o in un campo DATA

Indice: incrementato da una funzione FOR...NEXT

Lista: inizializzato da DIM e/o READ...DATA

Variabile: valore indeterminato, utilizzabile in più punti del programma, in condizioni diverse di elaborazione.

ORIGINE

Tastiera, programma, cassetta, dischetto.

LUNGHEZZA

Precisare il campo degli interi e decimali	2 + 2
intero	2

OSSERVAZIONI

Precisare il ruolo della variabile, i differenti valori contenuti.

DATA	NOME DEL PROGRAMMA:	PAGINA

IDENTIFICAZIONE DELLE VARIABILI

NOME SIMBOLICO	TIPO(*)	ORIGINE: TASTIERA	ORIGINE: PROGRAMMA	ORIGINE: CASSETTA	ORIGINE: DISCHETTO	LUNG.	OSSERVAZIONI

(*) LEGENDA: decisione, contatore, costante, puntatore, indice, lista, variabile.

TAVOLA SINOTTICA DELLE VARIABILI CON LE FUNZIONI DI INIZIALIZZAZIONE E DI RIMESSA ALLO STATO INIZIALE

TIPO DI VARIABILE	INIZIAL.	RIMESSA ALLO STATO INIZ.
DECISIONE	SI'	SI'
CONTATORE	FACOLTATIVO	FACOLTATIVO
COSTANTE	SI'	NO
PUNTATORE	SI	FACOLTATIVO
INDICE	SI'	FACOLTATIVO
LISTA	SI'	SI' (RESTORE)
VARIABILE	FACOLTATIVO	FACOLTATIVO

I FOGLI DI PROGRAMMAZIONE

Hanno un triplice uso:

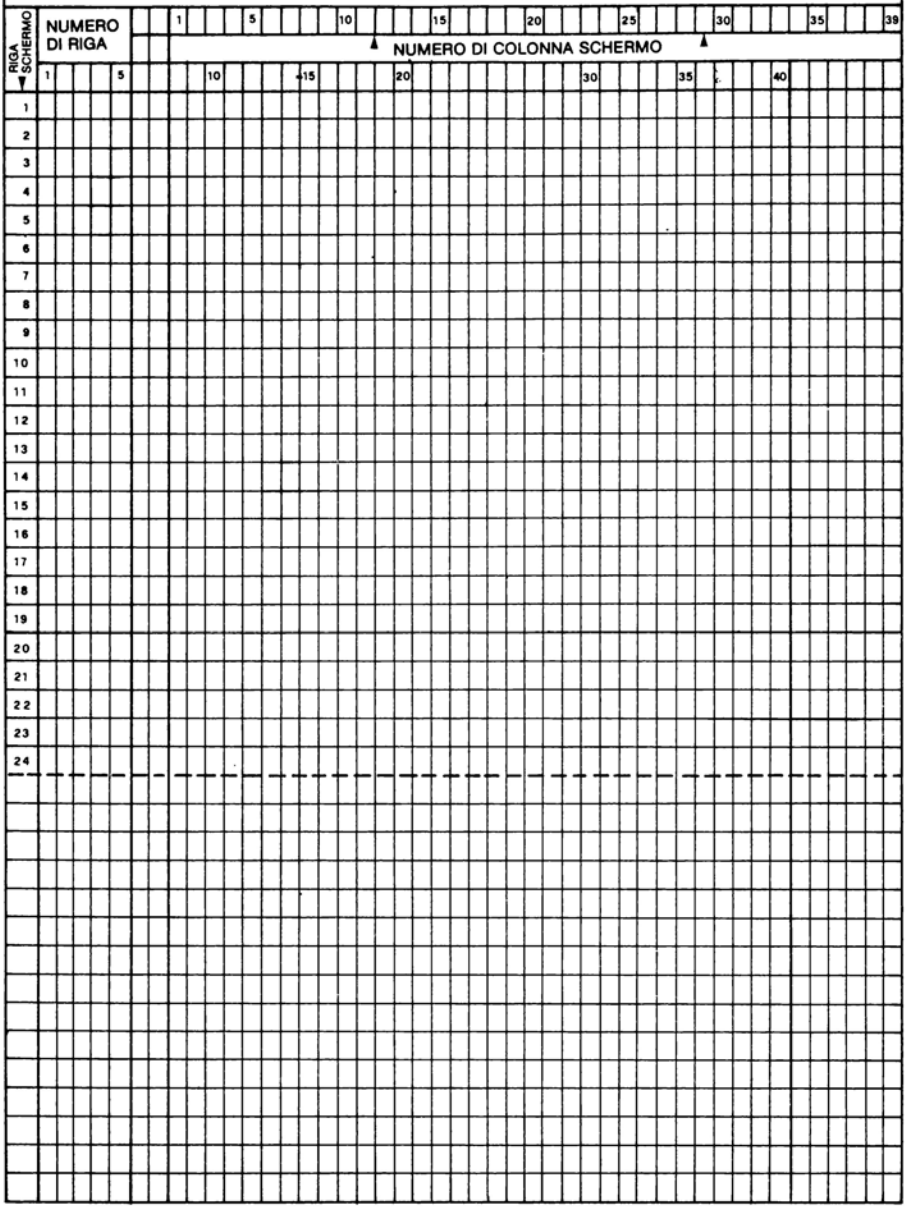
— Stabilire le maschere di schermo;
— Registrare righe delle istruzioni;
— Fornire le ascisse e ordinate di ciascun punto dello schermo.

ALLEGATI

— Controllo di acquisizione;
— Controllo di validità;
— Controllo di plausibilità;
— Controllo di compatibiltà interna (confronto delle variabili interne al programma);
— Controllo di compatibilità esterna (confronto delle variabili interne con quelle relative ai file di ingresso);
— Lista dei messaggi d'errore;
— Sviluppare la lista dei diversi messaggi d'errore. Comporre i messaggi d'errore nella stessa sequenza dei blocchi del programma, se possibile.
— Problemi di manutenzione. Elenco delle variabili, e del loro blocco d'uso relativamente a quelle che possono essere modificate durante l'elaborazione.

DATA	NOME DEL PROGRAMMA:	PAGINA

FOGLIO DI PROGRAMMAZIONE BASIC



Reproduction interdite © Editions du P.S.I

CAPITOLO 6

ESEMPIO DI REALIZZAZIONE

ESPOSIZIONE DEL PROBLEMA: DI CHE COSA SI TRATTA?

Di contabilizzare certe forniture di sci (sci, attacchi, scarponi). Gli ordini sono effettuati da diversi negozi specializzati nella vendita di questi prodotti. Ogni negozio rappresenta un cliente.

Per ogni cliente che ha effettuato un ordine, si calcola e si compone sullo schermo una fattura.

Periodicità di elaborazione	= secondo la richiesta
Volume delle transazioni	= minimo 1 fattura; massimo 10 fatture per sessione
Tempo di consegna dei risultati	= 24 ore dopo l'elaborazione, invio delle fatture a giro di posta.

Notiamo per inciso che partendo da un esempio semplice il nostro scopo è di dare importanza al metodo.

Prima di proseguire nello studio, facciamo una pausa di riflessione e domandiamoci se l'elaboratore di cui disponiamo è in grado di fornire le prestazioni richieste.

A priori, un elaboratore personale può reggere il volume delle transazioni da elaborare e rispondere nel tempo voluto, ma chiudiamo l'osservazione.

Il problema scelto è volutamente semplificato allo scopo di facilitare la dimostrazione.

Questa guida, essendo uno strumento di riflessione, può e deve essere un trampolino che permetta la realizzazione di progetti tecnici molto più complessi, nel campo dell'informatica applicata.

La qualità del contenuto di questi progetti dipenderà, per ognuno di noi, dalle facoltà di adattamento nell'utilizzare tale strumento e, naturalmente, dal saper trarre professionalmente profitto dall'esperienza.

COME TRATTARE IL PROBLEMA?

Il file "clienti" relativo a 10 negozi viene posto in memoria principale.

Il programma domanda all'inizio della sessione la data di fatturazione, comune a tutte le fatture, e l'ultimo numero di fattura immesso (relativo all'elaborazione precedente); se è la prima volta si deve immettere il valore zero.

Il programma si incarica in seguito di incrementarlo di +1 per ogni nuova fattura.

Per ogni nuova fattura da comporre, si devono immettere le informazioni seguenti, preferibilmente alla tastiera.

Il numero codice cliente

Permette di ricercare nel file "clienti" i dati di identificazione del relativo negozio, valore compreso tra 1 e 10.

La quantità degli articoli ordinati

Introdurre 0 se la quantità è nulla. In tal caso il progamma non elaborerà la riga corrispondente.

Una conferma dei dati immessi è richiesta dal programma.

Successivamente si passa al calcolo e alla composizione della fattura. E' previsto un sottoprogramma di formattazione degli importi prima di essere stampati.

Questa sottoprogramma è fondamentale per gli elaboratori personali non muniti di PRINT-USING.

Ultimata la fattura, il programma domanda se è l'ultima.

Se la risposta è negativa, il programma ritorna nella sequenza d'immissione di un nuovo codice cliente.

Se la risposta è affermativa, il programma invia un messaggio con significato opposto:

```
"PROGRAMMA TERMINATO"
READY
```

NOTE

Per non presentare un programma troppo complesso e appesantito da istruzioni, abbiamo volutamente scartato certe funzioni come:

— controllo dei dati immessi alla tastiera;
— sottoprogramma di cancellazione parziale dello schermo;
— sottoprogramma di stampa dei messaggi d'errore;
— scrittura della fattura su stampante;
— scrittura dell'ultimo numero di fattura immesso su file.

DIAGRAMMA DI FLUSSO GENERALE DEL PROGRAMMA

Prima di scrivere una sola riga di programma raccomandiamo di iniziare con la progettazione delle strutture generali, allo stesso modo di un architetto che disegna il progetto prima di iniziarne l'esecuzione.

Il diagramma di flusso permette di fare l'inventario di tutte le funzioni necessarie, rivela altresì quelle che possono mancare quando si prendono in considerazione tutte le diramazioni possibili del programma.

L'esempio presentato è piuttosto lineare e quindi abbastanza semplice da capire; i connettori A e B sono utili quando il diagramma di flusso si estende su diverse pagine.

Prima di andare avanti nella realizzazione del programma, analizziamo i dati da elaborare ordinandoli in modo logico con l'aiuto di alcuni concetti della matematica.

Affrontiamo il problema dal punto di vista della teoria degli insiemi. I concetti di insieme e sottoinsieme associati ai blocchi di elaborazione appariranno chiaramente.

— Nei blocchi "Ingresso" i dati sono: data del giorno, numero di fattura, numero di codice cliente...

— Nei blocchi di calcolo e di uscita altri dati sono determinati.

Il diagramma di flusso generale definisce 5 gruppi principali di dati.

1 - il gruppo dei dati di ingresso
2 - il gruppo delle costanti
3 - il gruppo dei dati calcolati
4 - il gruppo dei dati di transito (temporanei)
5 - il gruppo dei dati di uscita.

L'insieme dei dati da elaborare si estende dai sottoinsiemi logici a uno o più elementi.

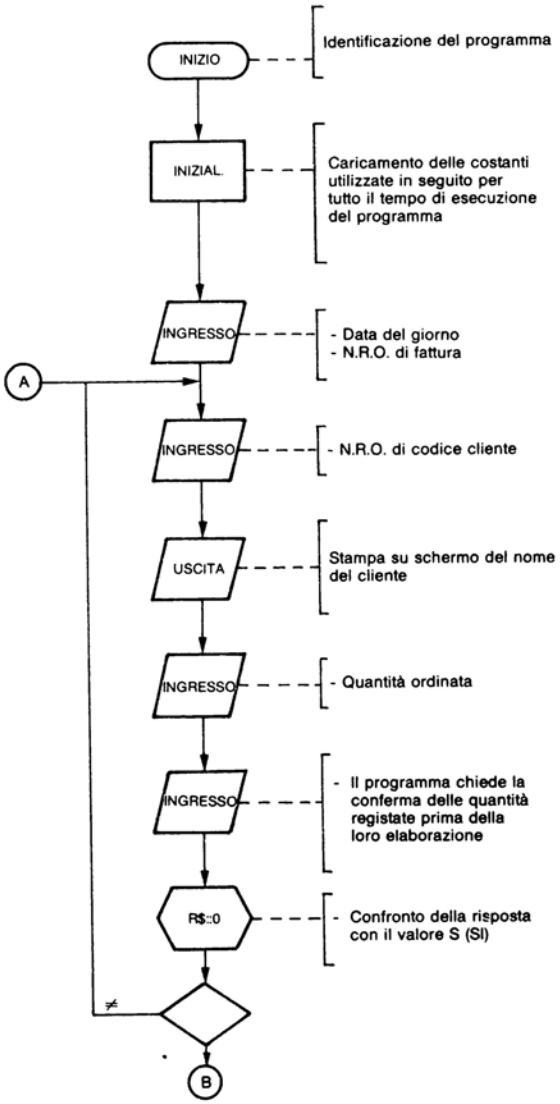
1 - L'insieme E1 dei dati d'ingresso:

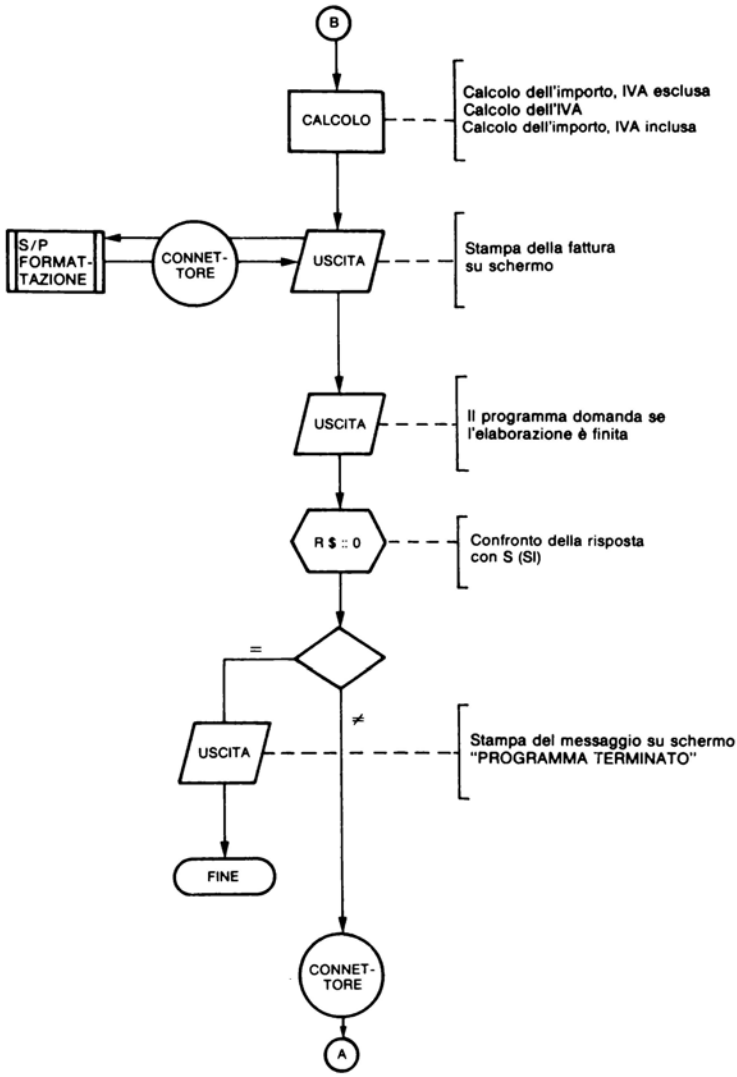
— la data di fatturazione (D$); immessa all'inizio della sessione, è utilizzata come "costante" in tutte le fatture prodotte.
— l'ultimo numero di fattura registrato in occasione della precedente sessione (F1); immesso all'inizio della sessione, viene incrementato di + 1 dal programma per ogni transazione.
— il numero di codice cliente (C1); immesso per ogni nuova transazione, serve come puntatore di ricerca nel file clienti registrato in memoria principale.
— la quantità di articoli ordinati Q1 per il numero di sci, Q2 per il numero di attacchi, Q3 per il numero di coppie di scarponi; dati immessi per ogni nuova transazione.

In breve gli elementi appartenenti all'insieme E1 sono:
(D$, F1, C1, Q1, Q2, Q3).

2 - L'insieme E2 dei dati memorizzati come costanti nella memoria principale:

— il file dei clienti riguardanti la consegna degli articoli ordinati (FC); nella maggior parte dei casi questi dati sono memorizzati su una periferica di grande capacità, quale il disco o nastro magnetico.
— il prezzo unitario di ciascun articolo, (P1) per il valore di una coppia di sci, (P2) per il valore di una coppia di attacchi, (P3) per il valore di una coppia di scarponi.


INIZIO
Identificazione del programma
INIZIAL.
Caricamento delle costanti utilizzate in seguito per tutto il tempo di esecuzione del programma
INGRESSO
- Data del giorno
- N.R.O. di fattura
A
INGRESSO
- N.R.O. di codice cliente
USCITA
Stampa su schermo del nome del cliente
INGRESSO
- Quantità ordinata
INGRESSO
- Il programma chiede la conferma delle quantità registate prima della loro elaborazione
R$::0
- Confronto della risposta con il valore S (SI)
≠
B


B
CALCOLO
Calcolo dell'importo, IVA esclusa
Calcolo dell'IVA
Calcolo dell'importo, IVA inclusa
S/P FORMAT-TAZIONE
CONNET-TORE
USCITA
Stampa della fattura su schermo
USCITA
Il programma domanda se l'elaborazione è finita
R$ :: 0
Confronto della risposta con S (SI)
=
≠
USCITA
Stampa del messaggio su schermo "PROGRAMMA TERMINATO"
FINE
CONNET-TORE
A

- il tasso dell'I.V.A. (T1)

In breve, gli elementi appartenenti all'insieme E2 sono:
(FC, P1, P2, P3, T1).

3 - L'insieme E3 dei dati calcolati dall'elaboratore:

- il numero della fattura F2
- il prezzo totale delle coppie di sci D1
- il prezzo totale degli attacchi D2
- il prezzo totale degli scarponi D3
- l'importo della fattura IVA esclusa .....T
- l'importo dell'IVA T2
- l'importo della fattura IVA inclusa T3

In breve, gli elementi appartenenti all'insieme E3 sono:
(F2, D1, D2, D3, T, T2, T3).

4 - L'insieme E4 dei dati ausiliari necessari per l'elaborazione dei dati d'ingresso, memorizzati come costanti e calcolati:

- l'area di lavoro temporanea per la registrazione del nome del cliente; N$

- la tabella contenente il file cliente; L$

- l'indice utilizzato per il funzionamento dei cicli iterativi; I

- le variabili utilizzate per i sottoprogrammi di formattazione; (S, X$, L$, U, J, Y, Y$).

In breve, gli elementi appartenenti all'insieme E4 sono:
(L$, N$, I, S, LS, U, J, Y, Y$).

5 - L'insieme E5 dei dati di uscita utilizzati per la composizione dei diversi messaggi conversazionali, maschera di acquisizione e di fatturazione che implicano i seguenti elementi:
(C1, N$, F2, D$, D1, D2, D3, T1, T2, T3, X$).

Effettuiamo ora le operazioni booleane negli insiemi definiti.

1 - Intersezione degli insiemi E3 e E5

E3

	D1	D2	D3		F2		T		T2	T3	
0	1	1	1	0	1	0	1	0	1	1	0

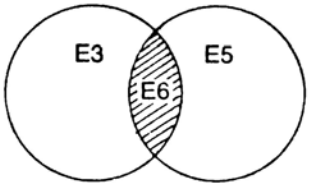
E5

C1	D1	D2	D3	D$	F2	N$	T	T1	T2	T3	X$
1	1	1	1	1	1	1	1	1	1	1	1

Determiniamo il sottoinsieme contenente gli elementi appartenenti a E3 e E5

E6

D1	D2	D3		F2		T		T2	T3
1	1	1		1		1		1	1

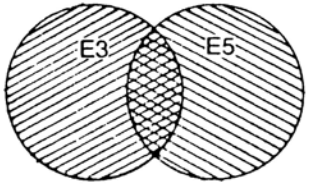


operazione "AND" logico

2 - Unione degli insiemi E3 e E4

E7

C1	D1	D2	D3	D$	F2	N$	T	T1	T2	T3	X$
1	1	1	1	1	1	1	1	1	1	1	1



operazione "OR" inclusivo

3 - Esclusione degli insiemi E2 e E5

E2

C1	D1	D2	D3	D$	F2	N$	T	T1	T2	T3	X$	FC	P1	P2	P3
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

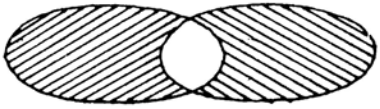
E5

C1	D1	D2	D3	D$	F2	N$	T	T1	T2	T3	X$	FC	P1	P2	P3
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Il sottoinsieme E8 presentato qui sotto contiene gli elementi appartenenti sia a E2, sia a E5, ma non a E2 e a E5 contemporaneamente.

E8

C1	D1	D2	D3	D$	F2	N$	T		T2	T3	X$	FC	P1	P2	P3
1	1	1	1	1	1	1	1		1	1	1	1	1	1	1



operazione "OR" esclusivo

Esempio di utilizzo delle tavole di verità per risolvere le condizioni complesse di elaborazione.

Riprendiamo perciò la sequenza di selezione dei vagoni, introdotta per illustrare il ruolo del simbolo di decisione, a pagina 32.

Nel diagramma di flusso si notano le condizioni multiple per andare sulle strade 2, 5, 4, 7 e 8. E con l'aiuto delle due tavole di verità il problema appare chiaro.

Tavola n. 1

Milano · $\overline{\text{Bologna}}$	andare alla tavola n. 2
$\overline{\text{Milano}}$ · Bologna	andare alla tavola n. 3
$\overline{\text{Milano}}$ · $\overline{\text{Bologna}}$	andare in sequenza (1)

(1) Se il codice direzione non corrisponde a quello di MI o BO, non esistono condizioni multiple, la scelta della strada dipende soltanto da un valore che è legato unicamente al codice destinazione.

Tavola n. 2 Smistamento dei vagoni per Milano

$M \cdot \overline{V}$	andare sulla strada 2
$\overline{M} \cdot V$	andare sulla strada 5
$\overline{M} \cdot \overline{V}$	andare sulla strada 8

Tavola n. 3 Smistamento dei vagoni per Bologna

$M \cdot \overline{V}$	andare sulla strada 7
$\overline{M} \cdot V$	andare sulla strada 4
$\overline{M} \cdot \overline{V}$	andare sulla strada 8

Proseguiamo ora nel seguito di questo capitolo.

MASCHERE DI SCHERMO

Le maschere di schermo sono delle fotografie dell'informazione elaborata. E' indispensabile disegnarle prima di realizzarle in programmazione.

Quando tutti i disegni sono stabiliti, basta scrivere le istruzioni. A questo livello la programmazione diventa una "peripezia". Costruito l'ambiente, il film può svolgersi. Silenzio, si gira! Motore.

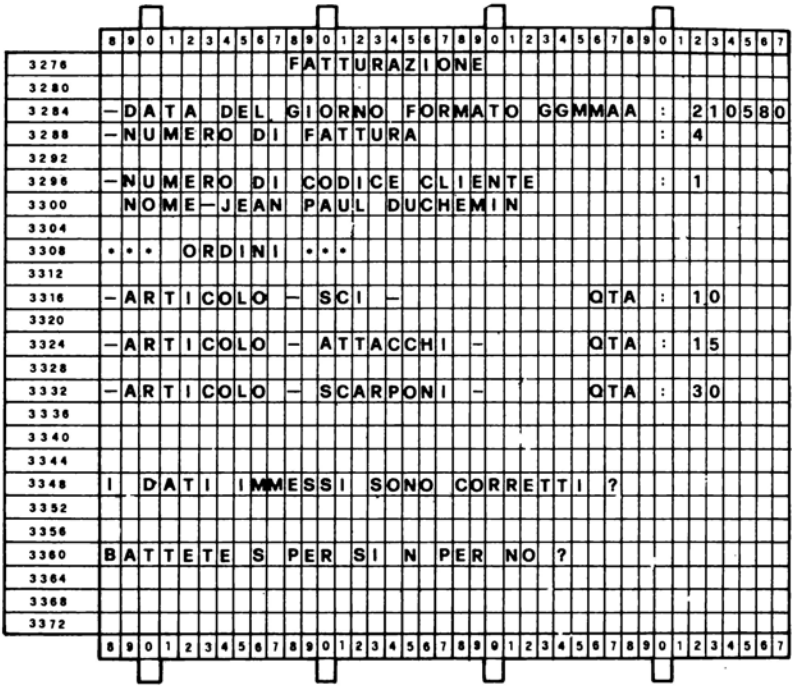
Nel caso della fatturazione sono richiesti due maschere (pagine 57 e 58). La prima serve per l'immissione dei dati indispensabili al calcolo di una fattura. Prima di passare alla maschera successiva, il programma domanda di confermare i valori immessi. In caso di errore, il programma ritorna nella sequenza d'immissione.

La seconda maschera rappresenta una fattura completa di un cliente.

Il numero del cliente permette di ricercare e di comporre automaticamente la ragione sociale, per mezzo di un file indicizzato, memorizzato nel programma nel campo DATA.

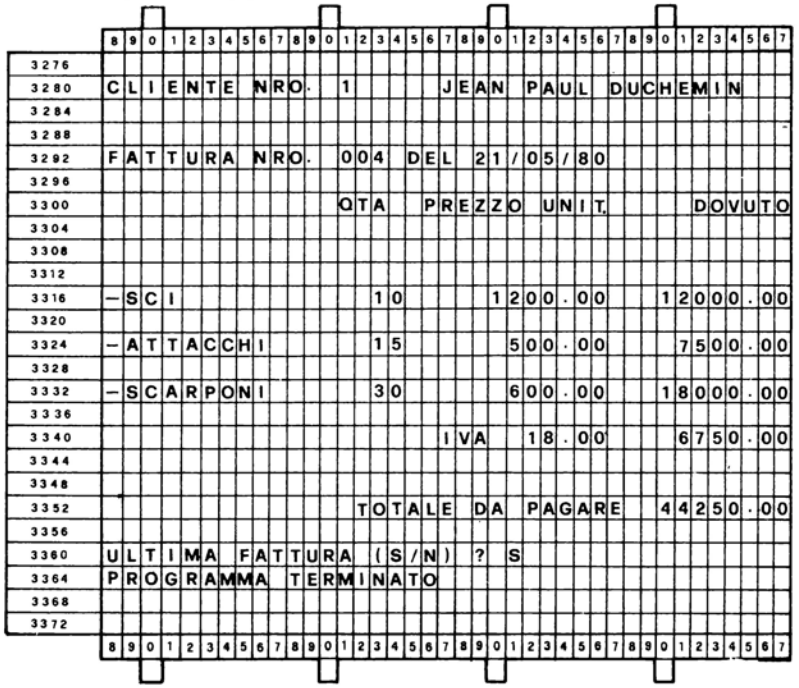
MASCHERA 1

GRIGLIA DI SCHERMO



MASCHERA 2

GRIGLIA DI SCHERMO

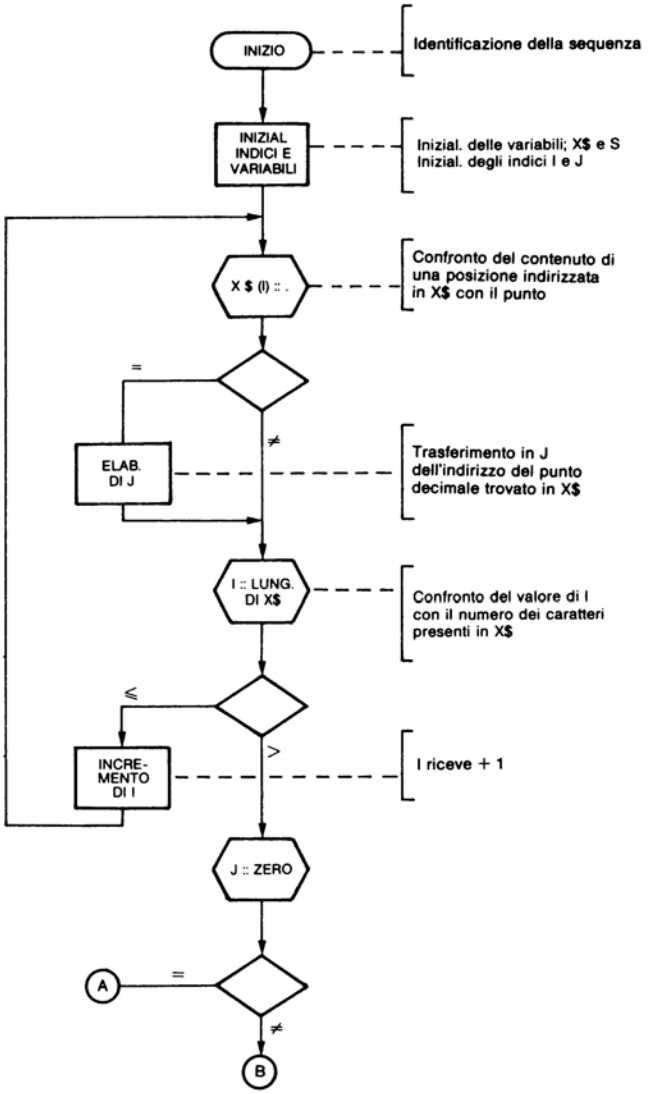


DIAGRAMMI DI FLUSSI DETTAGLIATI

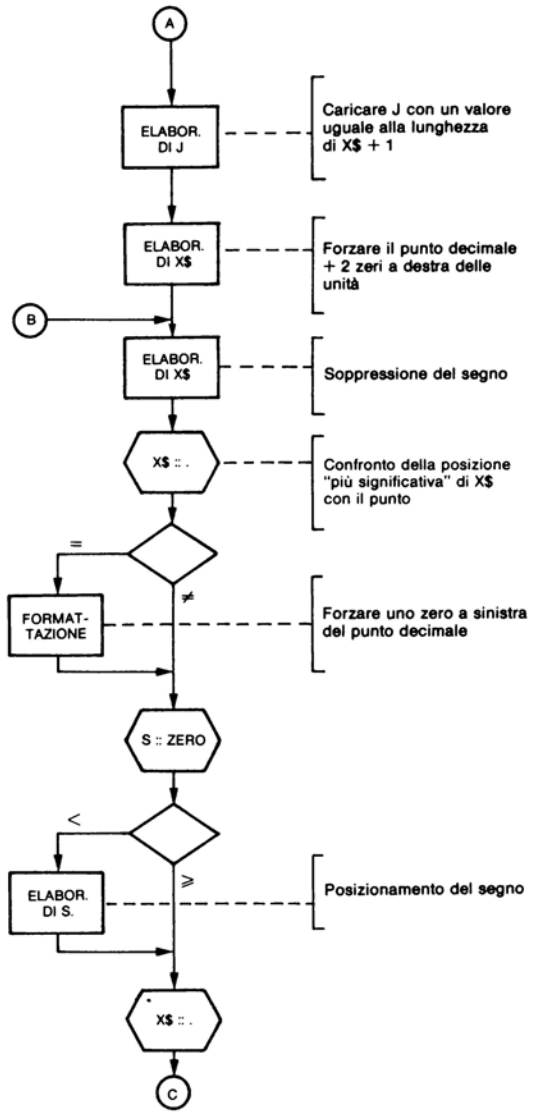
Si tratta, una funzione dopo l'altra, di dedurre un diagramma di flusso a un livello di precisione pari alla complessità del programma.

Il sottoprogramma di formattazione degli importi è una sequenza complessa, sempre interessante da vedere nei suoi dettagli. Il diagramma di flusso è, ripetiamolo, un eccellente strumento in questo campo per ispezionare e "smontare" i meccanismi più sottili.

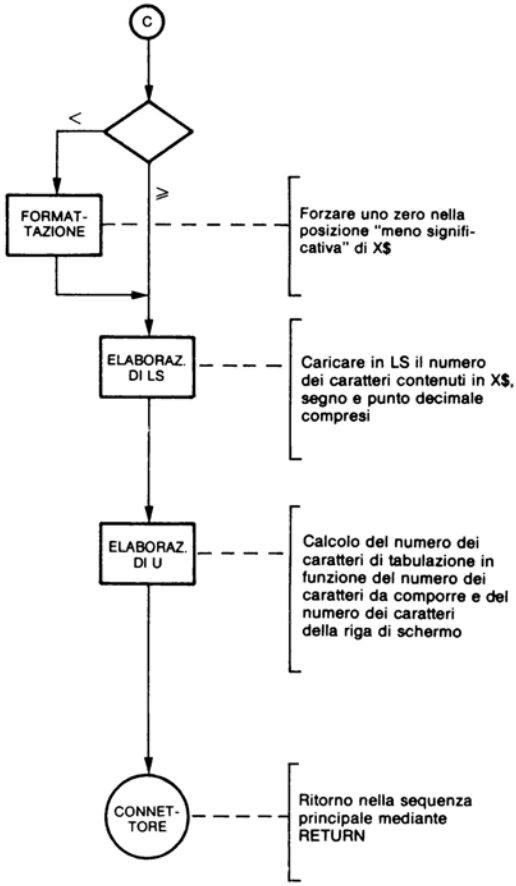
DATA	NOME DEL PROGRAMMA: **FATTURAZIONE** **S/P DI FORMATTAZIONE DEGLI IMPORTI**	PAGINA



DATA	NOME DEL PROGRAMMA **FATTURAZIONE** **S/P DI FORMATTAZIONE DEGLI IMPORTI**	PAGINA



DATA	NOME DEL PROGRAMMA: **FATTURAZIONE** **S/P DI FORMATTAZIONE DEGLI IMPORTI**	PAGINA



LA PROGRAMMAZIONE

È grande la tentazione di programmare direttamente un elaboratore personale. Ma la stesura preventiva del programma e l'utilizzo di un supporto strutturato e idoneo all'hardware scelto migliorano sensibilmente il lavoro del programmatore.

Sacrificare un'ora del proprio tempo per scrivere le sequenze su tale supporto, può rendere più spedita la messa a punto del programma (cinque volte di più, almeno). Inoltre, per gli utenti degli elaboratori personali che non hanno una stampante, la scrittura preventiva dei programmi, non solo è consigliabile, ma risulta indispensabile allorchè si affrontano programmi con più di una dozzina di righe.

FATTURAZIONE

FOGLIO DI PROGRAMMAZIONE **BASIC**

RIGA SCHERMO	NUMERO DI RIGA	
1	1100	REM • S/P DI FORMATTAZ. PER LA COMPOS
2	1110	REM • DEGLI IMPORTI
3	1120	J=0:REM • INIZ. INDICE J
4	1130	S=INT(S•100+.5)/100
5	1140	X$=STR$(S)
6	1150	FOR I=1TOLEN(X$)
7	1160	IF MID$(XS,I,1)="."THEN J=I
8	1170	NEXT I
9	1180	IF J=0 THEN J=LEN(X$)+1:X$=X$+".00"
10	1190	X$=MID$(X$,2,J+2):REM • CENTRATURA
11	1200	REM • DI X$ SOPPRIMENDO IL SEGNO
12	1210	IF MID$(X$,1,1)="."THEN X$=RIGHT$(MI$,1
13		)+X$
14		
15	1230	IF S<0 THEN X$="-"+X$
16	1240	IF MID$(X$)LEN(X$)-1,1)="."THENX$=X$+"0
17		"
18	1250	LS=LEN(X$)
19	1260	U=39-LS
20		
21		
22	1290	RETURN
23		
24		

CLASSIFICAZIONE DELLE VARIABILI

Si tratta, in qualche modo, di classificare le variabili utilizzate in un dato programma.

Nel programma di fatturazione sono state recensite 27 variabili.

Stendere l'elenco delle aree di lavoro di un programma può risultare utile, a più ragioni: da una parte la messa a punto è più rapida, e dall'altra le modifiche al programma, piccole o grandi che siano rispetto al progetto iniziale, sono largamente facilitate mediante il controllo del ruolo tenuto da ogni variabile.

I criteri di classificazione introdotti sono abbastanza evidenti per considerare questo supporto come un valido mezzo d'aiuto per la gestione di un programma.

DATA	NOME DEL PROGRAMMA: **FATTURAZIONE**	PAGINA

IDENTIFICAZIONE DELLE VARIABILI

NOME SIMBOLICO	TIPO(*)	ORIGINE: TASTIERA	PROGRAMMA	CASSETTA	DISCHETTO	LUNG.	OSSERVAZIONI
D$	Variabile	x				6	Data del giorno per il complesso delle fatture
F1	«	x				⩽3	N.ro di fattura, la prima di una serie
F2	«		x			⩽3	N.ro di fattura calcolato
C1	«	x				⩽2	N.ro di codice cliente
Q1	«	x				⩽3	Numero di articoli di sci
Q2	«	x				⩽3	« attacchi
Q3	«	x				⩽3	« scarponi
P1	Costante		x			⩽7	Prezzo unitario dell'articolo sci
P2	«		x			⩽7	« attacchi
P3	«		x			⩽7	« scarponi
D1	Variabile		x			⩽8	Totale Q1 x P1
D2	«		x			⩽8	Totale Q2 x P2
D3	«		x			⩽8	Totale Q3 x P3
T1	Costante		x			⩽4	IVA (tasso)
T2	Variabile		x			⩽7	Importo dell'IVA
T3	«		x			⩽8	Importo totale della fattura
R$	«	x				⩽1	Risposta SI o NO
N$	«		x			⩽30	Nome del cliente
L$	Lista		x				File dei clienti in DATA
I	Indice		x			⩽2	Loop FOR/NEXT
S	Variabile		x			⩽8	Area di lavoro per il S/P di formattazione
X$	Variabile		x			⩽8	Area di lavoro per il S/P di formattazione

(*) LEGENDA: decisione, contatore, costante, puntatore, indice, lista, variabile.

DATA	NOME DEL PROGRAMMA:	PAGINA

IDENTIFICAZIONE DELLE VARIABILI

NOME SIMBOLICO	TIPO(*)	ORIGINE: TASTIERA	ORIGINE: PROGRAMMA	ORIGINE: CASSETTA	ORIGINE: DISCHETTO	LUNG.	OSSERVAZIONI
L S	Variabile		x			1	Contiene la lunghezza di X$
U			x			⩽2	Numero dei caratteri da tabulare
J			x			1	Area di lavoro per il S/P di formattazione
Y $			x			⩽8	Area di lavoro per la stampa di Q1, Q2, Q3
Y			x			1	Numero di caratteri da tabulare

(*) LEGENDA: decisione, contatore, costante, puntatore, indice, lista, variabile.

PRESENTAZIONE DEL LISTATO DEL PROGRAMMA

Il listato completo del programma di fatturazione presentato nelle pagine seguenti comprende alcune note di commento inserite in ogni funzione principale, attraverso l'istruzione Basic - REM -; che permette di impaginare il listato per paragrafi; ciascuno di questi rappresenta una unità logica di elaborazione, fornendo una migliore comprensione del problema trattato.

La rilettura ne risulta quindi agevolata, soprattutto nelle fasi successive della messa a punto. Le ulteriori manutenzioni (se necessarie) saranno ugualmente facilitate, con l'immediata identificazione dei blocchi da modificare.

La suddivisione delle sequenze di elaborazione del programma si presenta così:

Righe di programma	*Commenti*
5 - 80	Identificazione del programma
100 -120	Inizializzazione delle costanti
200 - 260	Immissione della data del giorno e del primo N.RO di fattura
270	Memorizzazione del primo N.RO di fattura
300 - 330	Immissione del N.RO del codice cliente
340 - 380	Lettura del file dei nomi e trasferimento del nome trovato in un'area di lavoro
390 - 400	Composizione del nome
500 - 560	Immissione dei comandi
600 - 680	Convalida dei dati immessi - Rimessa allo stato iniziale di L$
700 - 770	Calcolo della fattura
800 - 890	Stampa della fattura
900 - 1010	Sequenza di fine lavoro - Rimessa allo stato iniziale di L$
1100 - 1290	Sottoprogramma di formattazione degli importi
2000 - 2030	Implementazione del file dei nomi dei clienti.

CAPITOLO 7

LA MESSA A PUNTO E MANUTENZIONE

LA MODALITA' DELLA MESSA A PUNTO DI UN PROGRAMMA

Realizzato il manuale di programmazione, si può iniziare a mettere in macchina le sequenze di elaborazione.

Questa procedura deve essere realizzata per tappe. Ogni tappa rappresenta una o più sequenze, legate tra loro da una relazione logica funzionale.

ESEMPIO: Nel nostro programma di fatturazione, il controllo della composizione del nome del cliente in funzione del codice immesso alla tastiera, richiede la presenza in memoria delle seguenti sequenze:

— sequenza 300 - 400	immissione del codice cliente
— sequenza 340 - 380	lettura del file dei nomi dei clienti
— sequenza 390 - 400	stampa del nome
— sequenza 2000 - 2030	implementazione del file dei nomi dei clienti.

Alla fine di ogni prova, si deve pensare di salvare il programma su cassettta o su dischetto, prima di passare alla tappa successiva.

Ciò è una saggia precauzione, in grado di evitarci la perdita del lavoro fatto a causa di qualche interruzione di tensione, con possibile distruzione dei dati in memoria.

Per le sequenze di elaborazione il cui controllo completo richiede diverse sessioni per l'imissione dei dati, deve essere realizzato un piano di prove che simulino tutte le condizioni reali di lavoro.

Ad esempio, nel contesto del programma di fatturazione, il controllo della composizione della fattura può basarsi su una tabella di decisione a voci limitate, comprendente tutti i casi possibili d'imissione di comandi.

Se avete una stampante, si consiglia di fare un piano di prove. Le uscite visive delle maschere di schermo sono la garanzia di buon funzionamento del programma; in occasione di ulteriori manutenzioni, servono d'interfaccia logica tra la vecchia e la nuova versione del programma.

Costruiamo la tavola con i seguenti elementi:

1 - Enunciato delle condizioni: 3 elementi costituiti dalle coppie di sci, attacchi, coppie di scarponi.

2 - Le regole: 3 condizioni = 2^3 combinazioni, cioè 8 regole possibili di quantità da immettere;

3 - Enunciato delle azioni: 8 sessioni per l'immissione dei comandi relativi alle 8 regole di elaborazione.

		Le regole							
		0	1	2	3	4	5	6	7
Enunciato delle condizioni	Sei Attacchi Scarponi	no no no	si no no	si si no	si no si	no si no	no si si	no no si	si si si
Enunciato delle azioni	Sessione 1	X							
	Sessione 2		X						
	Sessione 3			X					
	Sessione 4				X				
	Sessione 5					X			
	Sessione 6						X		
	Sessione 7							X	
	Sessione 8								X

LA MANUTENZIONE DEL PROGRAMMA

La manutenzione del programma è un evento frequente e banale e riguarda tutte le applicazioni di elaborazione dati, le regole e valori dei quali dipendono da fenomeni esterni all'applicazione.

Ad esempio, il valore del tasso dell'IVA è soggetto a modifiche. Il principio stesso della sua applicazione può essere rimesso in discussione.

Anche la creazione di un supporto d'informazione adattato alle regole di manutenzione deve permettere di recensire preventivamente i dati e le funzioni del programma che possono essere modificati.

Il manuale di manutenzione del programma deve comprendere:

— Il manuale del programma;
— l'elenco delle funzioni e delle variabili prese in considerazione durante la manutenzione;
— i livelli d'intervento nel programma precisando i paragrafi e i numeri di riga;
— le modifiche effettivamente realizzate;
— il piano di prove ottenuto secondo le modalità già dette per la messa a punto dei programmi;
— l'aggiornamento del numero di versione e la data del nuovo programma prima che venga scritto su cassetta o dischetto.
— la ristampa di una lista completa del programma;
— l'archiviazione della lista e del piano di prove della versione precedente.

APPENDICE I

LA TEORIA DEGLI INSIEMI

RICHIAMI DEI CONCETTI

Una collezione di elementi distinti e definiti è chiamato - insieme

— la collezione degli studenti di una classe,
— la collezione dei francobolli italiani,
— la collezione delle variabili intere di un programma, ecc.

Un insieme può essere definito:

— da una lista completa dei suoi elementi, ognuno separato da virgole e compresi tra due parentesi graffe.

L'insieme delle variabili intere del programma, P1 = { A%, B%, T% }:

— da un criterio di appartenenza o da una legge del tutto generale in grado di determinare se un elemento appartiene o non appartiene all'insieme.

L'insieme delle variabili intere = { x/x è una variabile intera }.

L'elemento x appartenente a un insieme E è indicato dall'espressione: x ε E.

Un insieme può essere ordinato:

— quando gli elementi di un insieme sono classificati si dice che l'insieme è ordinato.

Un insieme vuoto può essere indicato da: ∅

— quando una variabile intera A% (unico elemento di un insieme) non ha ancora ricevuto informazioni; tuttavia per essa si è riservato posto nella memoria; si dice allora che:

A è vuoto; non ha alcun elemento significativo;
l'insieme { A% } = ∅

Un insieme può avere sottoinsiemi.
Consideriamo gli insiemi,

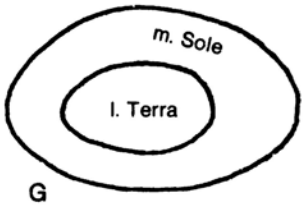
T = { x/x è in orbita intorno alla terra }
S = { x/x è in orbita intorno al sole }.

L'elemento che soddisfa i criteri di appartenenza "x è in orbita intorno alla terra" soddisfa anche il criterio di appartenenza "x è in orbita intorno al sole".

Pertanto un elemento qualunque di T è un sottoinsieme di S.

T è incluso in S ed è indicato da: T C S.

Si usano anche i cerchi di Eulero per rappresentare in modo più concreto queste nozioni.



I è un elemento della Terra
La Terra è un sottoinsieme del Sole

I ε T I appartiene all'insieme Terra
I ε S I appartiene all'insieme Sole

m ∉ T m non appartiene all'insieme Terra
m ε S m appartiene all'insieme Sole

G ∉ T G non appartiene all'insieme Terra
G ∉ S S non appartiene all'insieme Sole

Terra C Sole = Terra è inclusa nel Sole.

Un sottoinsieme può includere a sua volta altri sottoinsiemi.

Gli Italiani formano un sottoinsieme (I) della popolazione europea.

Gli Europei formano un sottoinsieme (EU) della popolazione terrestre.

La Terra è un sottoinsieme (TE) del sistema solare (SO).

Un sottoinsieme S di E può includere

qualche elemento di E	$S \subset E$
nessun elemento di E	$S = \varnothing$
tutti gli elementi di E	$S = E$
da cui	$\varnothing \subset E$
e	$E \subset E$

In altre parole, l'insieme vuoto e l'insieme stesso sono sempre sottoinsiemi di un insieme.

Un insieme può essere sempre considerato come un sottoinsieme di un insieme più vasto.

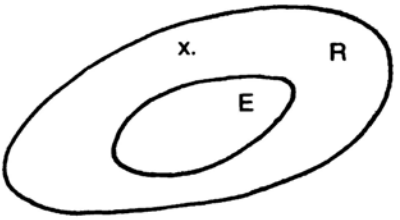
L'insieme che consideriamo più grande si dice insieme universale o insieme di riferimento. Ogni elemento esterno all'insieme universale può essere ignorato. Questo ci permette di risolvere i problemi specifici in un dato ambiente.

LE OPERAZIONI BOOLEANE

COMPLEMENTO DI UN INSIEME

Immaginiamo un insieme E in un insieme universale R e un elemento x appartenente a R ma non a E.

Si ha $x \varepsilon R$
$x \notin E$



L'elemento x appartiene a un insieme.
Questo insieme si chiama il complemento di E in R.

— Viene indicato con $\overline{E}$ o NOT E oppure $x \varepsilon \overline{E}$.

Il complemento di E è l'insieme di tutti gli elementi non appartenti ad E.

Gli elementi di R appartengono sia a E che a $\overline{E}$.

Da ciò, $E + \overline{E} = R$

Il complemento del complemento di E ($\overline{\overline{E}}$ - E doppia sbarra) è l'insieme degli elementi non appartenenti al complemento di E. I suoi elementi sono gli elementi di E.

Da cui, $\overline{\overline{E}} = E$

La nozione di complemento di un insieme è utilizzabile solamente dopo la definizione dell'insieme universale (R).

Ritroviamo l'applicazione del complemento di un insieme nel funzionamento di un circuito logico "NOT" di un microprocessore e così definito:

— circuito comprendente una porta di ingresso e una porta di uscita; ha la

proprietà di emettere un segnale di uscita in assenza del segnale d'ingresso, e di non emetterne in uscita se è presente uno d'ingresso.

INTERSEZIONE DI DUE INSIEMI

Regola: l'intersezione di 2 insiemi A e B è l'insieme di tutti gli elementi che appartengono contemporaneamente ad A e a B.

Operazione booleana "AND" logica, indicata da ∩:

	A ∩ B	A intesezione B
oppure	A e B	A AND B
	A · B	A e B

Esempio: consideriamo due sottoinsiemi A e B di un insieme universale R:

A =

a	b	c	d	e	f	g	h
1	0	1	0	1	0	1	1

B =

a	b	c	d	e	f	g	h
1	0	0	1	1	1	0	1

Determiniamo un sottoinsieme di R che contenga gli elementi comuni a A e a B e solo a questi; elementi in corrispondenza dei quali c'è 1 in A e 1 in B. Si ottiene così:

C =

a	b	c	d	e	f	g	h
1	0	0	0	1	0	0	1

$C = A \cap B$

In breve, l'operazione "AND" logico è definita dai quattro casi seguenti:

$0 . 0 = 0$
$0 . 1 = 0$
$1 . 0 = 0$
$1 . 1 = 1$

Ritroviamo l'applicazione dell'intersezione nel funzionamento di un circuito logico "AND" del microprocessore e così definito:

— circuito comprendente diverse porte d'ingresso e una porta d'uscita; ha la proprietà di emettere un segnale d'uscita solo se i segnali d'ingresso sono presenti su tutte le porte di ingresso.

UNIONE DI DUE INSIEMI

Regola: l'unione di due insiemi A e B è l'insieme degli elementi che appartengono

sia a A,
sia a B,
sia ai due insiemi.

Operazione booleana "OR" inclusivo, indicata con $\cup$:

$A \cup B$	(A unione B)
A o B	(A OR B)
A + B	(A o B)

Esempio. Riprendiamo i due sottoinsiemi A e B utilizzati nell'operazione booleana "AND" logico e vediamo quello che si ottiene nel sottoinsieme C, dopo l'operazione "OR" inclusivo.

C =

a	b	c	d	e	f	g	h
1	0	1	1	1	1	1	1

$C = A \cup B$

In breve, l'operazione "OR" inclusivo è definita dai quattro casi seguenti:

$0 . 0 = 0$
$0 . 1 = 1$
$1 . 0 = 1$
$1 . 1 = 1$

Ritroviamo l'applicazione dell'unione due insiemi nel funzionamento di un circuito logico "OR" inclusivo del microprocessore e così definito:

— circuito comprendente diverse porte d'ingresso e una porta d'uscita; ha la proprietà di emettere un segnale d'uscita a condizione che un segnale sia presente almeno su una delle porte d'ingresso.

ESCLUSIONE DI DUE INSIEMI

Regola: l'esclusione di 2 insiemi A e B è l'insieme degli elementi che appartengono

sia a A,
sia a B,

ma non contemporaneamente a A e B.
Operazione booleana "OR" esclusivo indicata da $\oplus$:

$A \oplus B$ (A o B incluso)
$A \cdot \overline{B} + \overline{A} \cdot B$

Esempio: Riprendiamo sempre i due sottoinsiemi A e B già visti nell'esempio dell'operazione booleana "AND" logico e vediamo quello che si ottiene nel sottoinsieme.

C =

a	b	c	d	e	f	g	h
0	0	1	1	0	1	1	0

$C = A \oplus B$

In breve, l'operazione "OR" esclusivo si riferisce ai quattro casi:

$0 \cdot 0 = 0$
$0 \cdot 1 = 1$
$1 \cdot 0 = 1$
$1 \cdot 1 = 1$

Ritroviamo l'applicazione dell'esclusione di due insiemi nel funzionamento di un circuito logico "OR" esclusivo del microprocessore e così definito:

— circuito comprendente due porte d'ingresso e una porta d'uscita; ha la proprietà di emettere un segnale di uscita a condizione che un segnale sia presente su una sola porta d'ingresso.

LE REGOLE DI PRIORITA' NELLE OPERAZIONI LOGICHE

La combinazione di diverse operazioni logiche booleane implica un ordine preciso a cui dobbiamo attenerci, se non vogliamo avere problemi di utilizzazione dei risultati.

Prendere in considerazione nell'ordine:

— le parentesi ( )
— i complementi NOT oppure −
— le intersezioni AND oppure .
— le unioni OR oppure +.

Prendiamo come esempio l'espressione: A + B . C
sarà eseguita per prima, successivamente l'operazione +.
Per cambiare l'ordine, dobbiamo far uso delle parentesi:

(A + B) . C

la parentesi sarà eseeguita per prima e . per ultimo.

Un altro esempio con: NOT A OR B, detto anche $\overline{A + B}$
l'ordine delle operazioni sarà: NOT e poi OR.

Invece, se scriviamo NOT (A OR B) detto anche $(\overline{A + B})$ l'ordine delle operazioni sarà: per prima le parentesi, poi OR e per ultimo NOT.

Teoremi di "De Morgan"

Questi teoremi forniscono alcune regole pratiche per ottenere il complemento d'una espressione.

Esempio:

$(\overline{A + B}) = \overline{A} \cdot \overline{B}$ il complemento dell'unione di più insiemi = l'intersezione dei loro complementi.

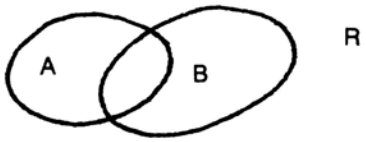
$(\overline{A \cdot B}) = \overline{A} + \overline{B}$ il complemento dell'intersezione di più insiemi = unione di più complementi.

$\overline{A \cap B} = \overline{A} \cup \overline{B}$
$\overline{A \cup B} = \overline{A} \cap \overline{B}$

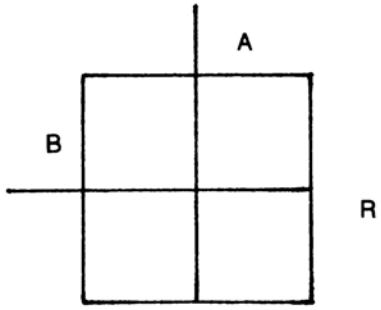
APPENDICE IV

DIAGRAMMA DI VEITCH

Possiamo rappresentare gli insiemi A e B in un insieme universale R mediante i cerchi di Eulero:



Possono essere rappresentati anche con un diagramma di Veitch.



I quattro piccoli quadrati rappresentano le 4 intersezioni $\overline{A} \cdot \overline{B}$, $\overline{A} \cdot B$, $A \cdot \overline{B}$, $A \cdot B$
Ogni intersezione è chiamata MINTERM.
Per ogni caso del diagramma di Veitch corrisponde un minterm.
Le dimensioni del diagramma rispondono alla seguente regola:

per 1 condizione si ha 2^1 combinazioni, cioè 2 casi
per 2 condizione si hanno 2^2 combinazioni, cioè 4 casi
per 3 condizione si hanno 2^3 combinazioni, cioè 8 casi
per 4 condizione si hanno 2^4 combinazioni, cioè 16 casi

Per motivi pratici non si consiglia andare oltre. L'insieme universale implicherebbe troppi elementi combinatori difficili da gestire.

Esempio d'utilizzazione.

Dobbiamo risolvere il seguente problema:

un risultato di calcolo deve essere addizionato in un accumulatore «TOTALE» da scegliere tra 4 accumulatori in funzione di determinate condizioni.

Chiamiamo i diversi accumulatori C1, C2, C3, C4.

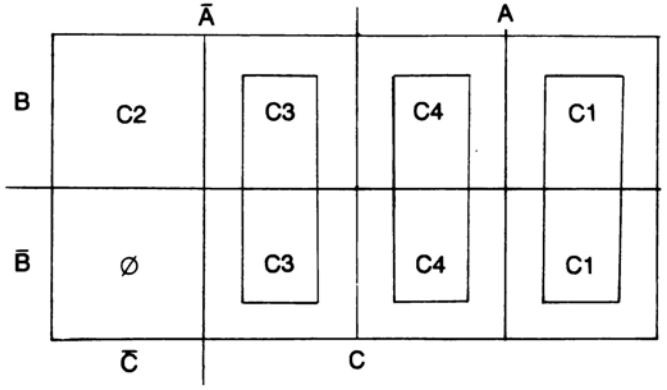
Chiamiamo le condizioni A, B, C.

deduciamo il diagramma applicando la regola seguente: per 3 condizioni si hanno 2^3 combinazioni, cioè 8 casi possibili (8 MINTERM).

Regola del programma:

— Accumulazione in C1 se $A \cdot \overline{B} \cdot \overline{C}$, $A \cdot B \cdot \overline{C}$
— Accumulazione in C2 se $\overline{A} \cdot B \cdot \overline{C}$
— Accumulazione in C3 se $\overline{A} \cdot \overline{B} \cdot C$, $\overline{A} \cdot B \cdot C$
— Accumulazione in C4 se $A \cdot \overline{B} \cdot C$, $A \cdot B \cdot C$

ciò provoca la formazione dei seguenti sottoinsiemi:



APPENDICE V

LE TAVOLE DI VERITA'

La tavola di verità è una tabella a doppia entrata in cui si trova la lista completa dei minterm formati dagli insiemi presi in considerazione (colonna di sinistra) così come la lista delle azioni da eseguire (colonna di destra). Riprendiamo l'esempio dell'invio del risultato di calcolo sugli accumulatori C1, C2, C3 o C4 e esprimiamo le condizioni per la somma mediante una tavola di verità.

$\bar{A} \cdot \bar{B} \cdot \bar{C}$	Nessun accumulo
$A \cdot \bar{B} \cdot \bar{C}$	Accumulazione in C1
$A \cdot B \cdot \bar{C}$	Accumulazione in C1
$\bar{A} \cdot \bar{B} \cdot C$	Accumulazione in C3
$\bar{A} \cdot B \cdot \bar{C}$	Accumulazione in C2
$A \cdot \bar{B} \cdot C$	Accumulazione in C4
$\bar{A} \cdot B \cdot C$	Accumulazione in C3
$A \cdot B \cdot C$	Accumulazione in C4

APPENDICE VI

PRESENTAZIONE DEL LISTATO DEL PROGRAMMA

```
10 REM ********************************
15 REM
20 REM        ESEMPIO  APPLICATIVO
25 REM        DELLA GUIDA  PRATICA
30 REM    "REALIZZAZIONE DEI PROGRAMMI"
35 REM
40 REM      FATTURAZIONE SEMPLIFICATA
45 REM      AUTORE : MICHEL BENELFOUL
50 REM
55 REM      COPYRIGHT  EDIZIONI P.S.I
60 REM            ED  AUTORE
65 REM
70 REM ********************************
75 REM
80 REM  ********************************
100 REM    INIZIALIZZAZIONE COSTANTI
101 REM ********************************
110 P1=1200:P2=500:P3=600
120 T1=18:M1$="10"
199 REM ********************************
200 REM IMMISSIONE DATA
201 REM E PRIMO NUMERO DI FATTURA
202 REM ********************************
220 PRINT;TAB(10);"FATTURAZIONE":PRINT
230 PRINT"DATA DEL GIORNO-FORMATO GGMMAA-";
240 INPUT D$
250 PRINT"NUMERO DI FATTURA";SPC(11);
260 INPUT F1
270 F2=F1:REM INIZ.CONTATORE N.RO FATT.
299 REM ********************************
300 REM IMMISSIONE N.RO CODICE CLIENTE
301 REM ********************************
310 PRINT
320 PRINT"NUMERO CODICE CLIENTE ";
330 INPUT C1
399 REM ********************************
400 REM RICERCA DEL NOME
401 REM NEL CAMPO DATA
402 REM ********************************
410 FOR I=1 TO 10
420 READ L$(I)
430 IF I=C1 THEN N$=L$(I)
```

```
440 NEXT I
450 PRINT:PRINT" CLIENTE ";N$;:PRINT
499 REM ******************************
500 REM IMMISSIONE ORDINI
501 REM ******************************
505 PRINT"+++ ORDINI +++":PRINT
510 PRINT"ARTICOLI         SCI          QTA";
520 INPUT Q1:PRINT
530 PRINT"ARTICOLI         ATTACCHI         QTA";
540 INPUT Q2:PRINT
550 PRINT"ARTICOLI         SCARPONI         QTA";
560 INPUT Q3:PRINT:PRINT:PRINT
599 REM ******************************
600 REM IL PROGRAMMA CHIEDE
601 REM LA CONFERMA
602 REM DEI DATI IMMESSI
603 REM ******************************
610 PRINT"I DATI IMMESSI SONO"
620 PRINT"CORRETTI?":PRINT
630 PRINT"RISPONDETE S PER SI N PER NO ";
640 INPUT R$
650 IF R$="S" THEN 700
660 RESTORE:GOTO 300:REM RITORNO NELLA SEQ.D'IMMISS.
699 REM ******************************
700 REM CALCOLO DELLA FATTURA
701 REM ******************************
710 D1=P1*Q1:D2=P2*Q2:D3=P3*Q3
720 T=D1+D2+D3:REM TOTALE IVA ESCL.
730 T2=T1*T/100:REM IVA
740 T3=T+T2:REM TOTALE DA PAGARE
750 F2=F2+1:REM N.RO FATTURA + 1
799 REM ******************************
800 REM STAMPA DELLA FATTURA
801 REM ******************************
810 PRINT"CLIENTE N.RO";C1;" ";N$:PRINT:PRINT
820 PRINT"FATTURA N.RO ";F2;" DEL";:PRINT LEFT$(D$,2);"/";
825 PRINT MID$(D$,3,2);"/";
830 PRINT RIGHT$(D$,2):PRINT
840 PRINT TAB(15);"QTA";SPC(4);"P.U.";SPC(6);"DOVUTO":PRINT:
    PRINT
845 S=D1:GOSUB 1100:Y$=STR$(Q1):Y=LEN(Y$):Y=18-Y
847 IF Q1<=0 THEN PRINT:GOTO 855
850 PRINT"SCI";TAB(Y) Q1;P1;TAB(U) X$:PRINT
855 S=D2:GOSUB 1100:Y$=STR$(Q2):Y=LEN(Y$):Y=18-Y
857 IF Q2<=0 THEN PRINT:GOTO 865
860 PRINT"ATTACCHI";TAB(Y) Q2;P2;TAB(U) X$:PRINT
865 S=D3:GOSUB 1100:Y$=STR$(Q3):Y=LEN(Y$):Y=18-Y
867 IF Q3<=0 THEN PRINT:GOTO 875
870 PRINT"SCARPONI";TAB(Y) Q3;P3;TAB(U) X$:PRINT
875 S=T2:GOSUB 1100
880 PRINT TAB(16);"IVA ";T1;TAB(U) X$
885 S=T3:GOSUB 1100:PRINT
890 PRINT TAB(13);"TOTALE DA PAGARE";TAB(U);X$:PRINT
899 REM ******************************
900 REM TEST ULTIMA FATTURA
901 REM ******************************
```

```
910 INPUT"ULTIMA FATTURA (S/N) ";R$
920 IF R$="S" THEN 1000
930 RESTORE:GOTO 300
999 REM ******************************
1000 REM FINE DEL PROGRAMMA
1001 REM ******************************
1010 PRINT"PROGRAMMA TERMINATO"
1020 END
1100 REM ******************************
1101 REM ROUTINE DI FORMATTAZIONE
1102 REM       DEGLI IMPORTI
1103 REM ******************************
1120 J=0:REM INIZ. INDICE
1130 S=INT(S*100+.5)/100
1140 X$=STR$(S)
1150 FOR I=1 TO LEN(X$)
1160 IF MID$(X$,I,1)="." THEN J=I
1170 NEXT I
1180 IF J=0 THEN J=LEN(X$)+1:X$=X$+".00"
1190 X$=MID$(X$,2,J+2):REM CENTRATURA
1200 REM DI X$ SOPPRIMENDO IL SEGNO
1210 IF MID$(X$,1,1)="." THEN X$=RIGHT$(M1$,1)+X$
1230 IF S<0 THEN X$="-"+X$
1240 IF MID$(X$,LEN(X$)-1,1)="."THEN X$=X$+"0"
1250 LS=LEN(X$)
1260 U=39-LS
1290 RETURN
1299 REM ******************************
2000 REM FILE CLIENTI
2002 REM ******************************
2010 REM I N.RI DI CODICE CLIENTE
2020 REM SONO LE 2 ULTIME
2030 REM CIFRE DEL N.RO DI RIGA
2040 REM ESEMPIO:CLIENTE 8=2108
2099 REM ******************************
2101 DATA JEAN PAUL DUCHEMIN
2102 DATA ARTHUR PERE & FILS
2103 DATA ETABLISSEMENT RIQUET
2104 DATA MAUDUIT SHOP
2105 DATA J.C.KILLY
2106 DATA PERILLAT CENTER
2107 DATA MERIBEL BABIOLES
2108 DATA COURCHEVEL VOG
2109 DATA SERRE-CHEVALIER IN
2110 DATA VAL D'ISERE COMPETTTION
```

Destinato agli utenti di personal con una certa pratica di programmazione BASIC, il libro propone un metodo per la realizzazione dei programmi.
Per smitizzare il "dialetto" dell'informatica comincia col fornire un glossario completo e rigoroso.
È poi affrontato il problema dell'analisi di un sistema, con particolare attenzione al metodo di scelta dell'hardware, alle possibili riorganizzazioni del sistema ed alle esigenze di elaborazione dei dati.
Col supporto di modelli basati sulla matematica moderna e sulla teoria degli insiemi si affronta infine il problema dell'organizzazione logica dei dati.
Il metodo proposto per la realizzazione dei programmi si articola in: definizione, studio delle soluzioni, programmazione, messa a punto e manutenzione.
Esempi completi e semplici illustrazioni accompagnano e facilitano il cammino del lettore.

L. 10.000 Cod. 401H ISBN 88-7056-039-2